



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

TRABAJO FIN DE GRADO

Grado en Ingeniería Biomédica

**CLASIFICACIÓN AUTOMÁTICA DE LINFOCITOS ANORMALES
PROCEDENTES DE LINFOMAS DE BAJA PREVALENCIA
UTILIZANDO FEW-SHOT LEARNING**



Memoria

Autor:	Marc Font Pons
Director:	José Julián Rodellar Benedé
Co-Director:	Edwin Santiago Alférez Baquero
Convocatoria:	Junio 2020

Resum

L'anàlisi de sang mitjançant frotis sanguinis constitueix una eina molt útil per a la detecció d'anomalies i patologies. El grup d'investigació CellsiLAB, resultat de la col·laboració de la Universitat Politècnica de Catalunya i l'Hospital Clínic, ha desenvolupat mètodes automàtics de classificació de cèl·lules sanguínies per a diferents malalties. És necessari també desenvolupar una eina de classificació per aquells limfomes que tenen una incidència o prevalença molt baixa i no es tenen dades suficients per tractar el problema com una classificació d'imatges convencional.

Aquest projecte presenta un marc teòric de la fisiologia i biologia necessàries per comprendre els cinc tipus de limfomes que es classificaran: el limfoma de Burkitt, el limfoma fol·licular, la tricoleucèmia (HCL), el limfoma de cèl·lules del mantell (LCM) i la leucèmia limfocítica crònica (LLC). S'estableixen també fonaments de xarxes neuronals i *Few-Shot Learning* (FSL).

S'utilitzaran les arquitectures ResNet34 i ResNet50 per establir una metodologia de classificació convencional on s'aconsegueixen unes exactituds del 93% i del 98% respectivament. S'utilitzaran xarxes neuronals siameses (SNN) per afrontar el problema des del punt de vista del FSL. Utilitzar una xarxa neuronal siamesa amb arquitectura ResNet34 preentrenada aconsegueix els millors resultats de classificació.

Paraules clau: Classificació automàtica, cèl·lules sanguínies, baixa incidència, limfoma de Burkitt, limfoma fol·licular, tricoleucèmia, limfoma de cèl·lules del mantell, leucèmia limfocítica crònica, xarxes neuronals siameses, ResNet, Few-Shot Learning.

Resumen

El análisis de sangre mediante frotis sanguíneos es una herramienta útil para la detección de anomalías y patologías. El grupo de investigación CellsiLAB, resultado de la colaboración de la Universitat Politècnica de Catalunya y el Hospital Clínic, ha desarrollado métodos automáticos de clasificación de células sanguíneas para distintas anomalías. Era necesario también desarrollar una herramienta de clasificación para aquellos linfomas con muy baja incidencia de los cuáles no se tiene una base de datos suficiente para afrontar el problema como una clasificación de imágenes convencional.

Este proyecto presenta un marco teórico de la fisiología y biología necesario para comprender los cinco tipos de linfomas que se clasificaran: el linfoma de Burkitt, el linfoma folicular, la tricoleucemia (HCL), el linfoma de células del manto (LCM) y la leucemia linfocítica crónica (LLC). Se establecen también fundamentos de redes neuronales y *Few-Shot Learning* (FSL).

Se utilizarán arquitecturas ResNet34 y ResNet50 para establecer una metodología de clasificación convencional dónde se consigue una exactitud de clasificación del 93% y del 98% respectivamente. Se utilizarán redes neuronales siamesas (SNN) para afrontar el problema desde el punto de vista del FSL. Utilizar una red neuronal siamesa con arquitectura ResNet34 pre-entrenada consigue los mejores resultados de clasificación.

Palabras clave: Clasificación automática, células sanguíneas, baja incidencia, linfoma de Burkitt, linfoma folicular, tricoleucemia, linfoma de células del manto, leucemia linfocítica crónica, redes neuronales siamesas, ResNet, Few-Shot Learning.

Abstract

Blood analysis using blood smears is a common and useful tool for disease and anomalies detection. The research group CellsiLAB, fruit of collaboration between the Universitat Politècnica de Catalunya and the Hospital Clínic, has been developing methods of automatic classification of blood cells for different pathologies. CellsiLAB needed the development of a classification tool for very low incidence lymphomas that do not have enough data to be dealt with as a conventional image classification.

This project presents the biology and physiology concepts needed to understand the five different types of lymphoma that will be classified: Burkitt's lymphoma, follicular lymphoma, hairy cell leukaemia (HCL), mantle cell lymphoma (LCM) and chronic lymphocytic leukaemia (LLC). Neural network and Few-Shot Learning (FSL) fundamentals are also established.

ResNet34 and ResNet50 architectures will be used to establish a conventional method of classification obtaining 93% and 98% accuracy results. Siamese Neural Networks (SNN) will be used to face this challenge from a FSL standpoint. Using a Siamese neural network with a pre-trained ResNet34 architecture yields the best classification results.

Keywords: Automatic classification, blood cells, low incidence, Burkitt's lymphoma, follicular lymphoma, hairy cell leukaemia, mantle cell lymphoma, chronic lymphocytic leukaemia, Siamese neural networks, ResNet, Few-Shot Learning.



Agraïments

En primer lloc m'agradaria donar les gràcies a les persones del grup de recerca CellsiLAB, en especial al meu director José Rodellar i a la Dra. Anna Merino per haver-me cedit les imatges, indispensables per la realització d'aquest TFG.

Al meu codirector Santiago, gràcies pels innumerables correus i reunions de Skype que m'ajudaven a solucionar cada dubte que m'anava sorgint.

Gràcies als meus companys de pis, en especial a la Marina. Gràcies per escoltar-me quan t'explicava tots els problemes que tenia per poc que t'interessessin.

I als de casa: papa, mama, Ferran; gràcies per tot.



Glosario

PHSC: Célula precursora pluripotencial

FDG-PET: Tomografía por emisión de positrones con 18F-fluorodeoxiglucosa

HCL: *Hairy Cell Leukaemia*, Tricoleucemia

LCM: Linfoma de células del manto

LLC: Leucemia linfocítica crónica

DL: *Deep Learning* – aprendizaje profundo

FSL: *Few-Shot Learning*

ML: *Machine Learning* – aprendizaje automático

ReLU: *Rectified Linear Unit*

NN: *Neural Network* – Red neuronal

CNN: Redes neuronales convolucionales

SNN: *Siamese Neural Network* – red neuronal siamesa

Sensor CCD: Dispositivo de carga acoplada

Fine tuning: Ajustes de entrenamiento de la red neuronal

PCA: *Principal Component Analysis* – Análisis de componentes principales

Índice

RESUM	I
RESUMEN	II
ABSTRACT	III
AGRAÏMENTS	V
GLOSARIO	VI
1. PREFACIO	1
1.1. Origen del trabajo	1
1.2. Motivación	1
1.3. Requisitos previos	1
2. INTRODUCCIÓN	3
2.1. Objetivos del trabajo.....	3
2.2. Alcance del trabajo	3
3. BASES BIOLÓGICAS	5
3.1. Sangre y células sanguíneas.....	5
3.1.1. Eritrocitos.....	6
3.1.2. Leucocitos	6
3.1.3. Plaquetas	10
3.1.4. Plasma.....	10
3.2. El sistema linfático	10
3.3. Estudio de la sangre	12
3.4. Trastornos de los leucocitos	13
3.4.1. Neutrófilos	14
3.4.2. Eosinófilos	14
3.4.3. Basófilos.....	15
3.4.4. Monocitos	15
3.4.5. Linfocitos.....	16
3.5. Introducción a linfomas	16
3.5.1. Linfoma de Burkitt	18
3.5.2. Tricoleucemia	19
3.5.3. Linfoma de células del manto	20
3.5.4. Linfoma folicular	21

3.5.5.	Leucemia linfocítica crónica.....	22
4.	FUNDAMENTOS DE REDES NEURONALES Y FEW-SHOT LEARNING	23
4.1.	Introducción a las redes neuronales	23
4.1.1.	Neuronas artificiales	23
4.1.2.	Funciones de activación	24
4.1.3.	Funciones de pérdida.....	26
4.1.4.	Métricas, underfitting y overfitting	26
4.1.5.	Tipos de capas	28
4.2.	ResNet.....	29
4.2.1.	Funcionamiento de la ResNet34.....	32
4.3.	Few-shot learning	36
4.4.	Red Neuronal Siamesa (SNN)	37
5.	ANÁLISIS DEL CONJUNTO DE DATOS Y METODOLOGÍA APLICADA	39
5.1.	Adquisición de las imágenes.....	39
5.2.	Análisis del conjunto de datos.....	40
5.3.	Metodología de trabajo.....	41
5.3.1.	Bloque 1 (ResNet34 convencional).....	42
5.3.2.	Bloque 2 (ResNet50 convencional).....	44
5.3.3.	Bloque 3 (SNN con arquitectura ResNet34 sin pre-entrenamiento)	45
5.3.4.	Bloque 4 (SNN con arquitectura ResNet50 sin pre-entrenamiento)	51
5.3.5.	Bloque 5 (SNN con arquitectura ResNet34 con pre-entrenamiento).....	52
5.3.6.	Bloque 6 (SNN con arquitectura ResNet50 con pre-entrenamiento).....	53
6.	RESULTADOS	55
6.1.	Resultados del modelo ResNet34 convencional (Bloque 1)	55
6.2.	Resultados del modelo ResNet50 convencional (Bloque 2)	60
6.3.	Comparativa ResNet34 y ResNet50 convencionales	65
6.4.	Resultados del modelo SNN (ResNet34) sin pre-entrenamiento (Bloque 3)	66
6.5.	Resultados del modelo SNN (ResNet50) sin pre-entrenamiento (Bloque 4)	72
6.6.	Resultados del modelo SNN (ResNet34) con pre-entrenamiento (Bloque 5).....	76
6.7.	Resultados del modelo SNN (ResNet50) con pre-entrenamiento (Bloque 6).....	81
6.8.	Comparativa SNN (ResNet34 y ResNet50) con y sin pre-entrenamiento	85
7.	ANÁLISIS DEL IMPACTO AMBIENTAL	89
8.	CONCLUSIONES Y EXTENSIONES FUTURAS	91

8.1.	Conclusiones	91
8.2.	Extensiones futuras	92
9.	ANÁLISIS ECONÓMICO	93
9.1.	Costes de hardware	93
9.2.	Costes de software	93
9.3.	Costes de personal	94
9.4.	Costes indirectos	94
9.5.	Presupuesto total	95
BIBLIOGRAFÍA		96

Índice de figuras

Figura 3.1. Formación de diferentes células sanguíneas a partir de la célula precursora pluripotencial (PHSC) en la médula ósea (3).	5
Figura 3.2. Tipos de leucocitos (9).	7
Figura 3.3. Granulocitos (10).	7
Figura 3.4. Génesis de los leucocitos. Las diferentes células de la serie mielocítica son: 1, el mieloblasto; 2, el promielocito; 3, el megacariocito; 4, el metamielocito neutrófilo; 5, el metamielocito neutrófilo joven; 6, el metamielocito neutrófilo “cayado”; 7, el neutrófilo polimorfonuclear; 8, el mielocito eosinófilo; 10, el eosinófilo polimorfonuclear; 11, el mielocito basófilo; 12, el basófilo polimorfonuclear; 13-16, estadios de formación del monocito (3).	8
Figura 3.5. Anatomía del sistema linfático (15).	11
Figura 3.6. Esquema de un ganglio linfático no neoplásico (18).	12
Figura 3.7. Frotis sanguíneo utilizando la tinción de May-Grünwald-Giemsa (20).	13
Figura 3.8. Células de Reed-Sternberg (36).	17
Figura 3.9. Célula linfoide anormal en SP presente en pacientes con linfoma de Burkitt.	19
Figura 3.10. Célula linfoide anormal en SP presente en pacientes con tricoleucemia.	19
Figura 3.11. Célula linfoide anormal en SP presente en pacientes con linfoma de células del manto.	20
Figura 3.12. Célula linfoide anormal en SP presente en pacientes con linfoma folicular.	21
Figura 3.13. Célula linfoide anormal en SP presente en pacientes con leucemia linfocítica crónica.	22
Figura 4.1. Neurona artificial (59).	23
Figura 4.2. Función de activación ReLU (63).	25

Figura 4.3. Función de activación sigmoide (63).	25
Figura 4.4. Función de activación tanh (63).	26
Figura 4.5. Ejemplo de matriz de confusión (66).	27
Figura 4.6. Efectos del underfitting y overfitting en un problema de clasificación (68).	27
Figura 4.7. Error de predicción de un modelo en función de su complejidad (69).	28
Figura 4.8. Diferentes capas de una red neuronal (58).	28
Figura 4.9. Error del modelo frente a la profundidad de la red neuronal (74).	30
Figura 4.10. Bloque residual (75).	30
Figura 4.11. Arquitecturas de una red neuronal de 34 capas normal (izquierda) y una ResNet34 (derecha) (75).	31
Figura 4.12. Visualización de la ResNet34 (76).	32
Figura 4.13. Conv1 – Convolución (76).	33
Figura 4.14. Conv1 – Pooling (76).	33
Figura 4.15. Layer1, bloque 1, operación 1 (76).	34
Figura 4.16. Layer1, bloque 1 (76).	34
Figura 4.17. Layer 1 (76).	35
Figura 4.18. Layer 2, bloque 1, operación 1 (76).	35
Figura 4.19. Projection Shortcut (76).	36
Figura 4.20. Layer 2, bloque 1 (76).	36

Figura 4.21. Layer 2 (76).	36
Figura 4.22. Arquitectura de una SNN (79).	37
Figura 5.1. CellaVision DM96 (84).	39
Figura 5.2. Diagrama para visualizar la metodología de trabajo.	41
Figura 5.3. Relación de las imágenes con su etiqueta de clase.	42
Figura 5.4. Sección de código 1: data y cnn_learner.	42
Figura 5.5. Sección de código 2: fit_one_cycle, ClassificationInterpretation y top_losses.	43
Figura 5.6. Función de pérdida vs LR del modelo del bloque 1.	44
Figura 5.7. Función de pérdida vs LR del modelo del bloque 2.	45
Figura 5.8. Primeras cinco filas del dataframe.	45
Figura 5.9. Primeras cinco filas del dataframe duplicado.	46
Figura 5.10. Sección de código 3: ItemBase y ImageList.	47
Figura 5.11. Imagen transformada.	48
Figura 5.12. Sección de código 4: SiameseNet, forward y get_embedding.	48
Figura 5.13. Sección de código 5: ContrastiveLoss.	49
Figura 5.14. Objetivo de la función de pérdida Contrastive Loss (97).	49
Figura 5.15. Función de pérdida vs LR del modelo del bloque 3.	50
Figura 5.16. Sección de código 6: Extract_embedding.	50
Figura 5.17. Ejemplo de entradas para la función oneshot.	51

Figura 5.18. Sección de código 7: oneshot.	51
Figura 5.19. Función de pérdida vs LR del modelo del bloque 4.	52
Figura 5.20. Sección de código 8: Guardado de pesos en un diccionario.	52
Figura 5.21. Función de pérdida vs LR del modelo del bloque 5.	53
Figura 5.22. Función de pérdida vs LR del modelo del bloque 6.	54
Figura 6.1. Top_losses de la clasificación inicial del bloque 1.	57
Figura 6.2. Top_losses de la segunda clasificación del bloque 1.	59
Figura 6.3. Show_results del bloque 1.	60
Figura 6.4. Top_losses de la clasificación inicial del bloque 2.	62
Figura 6.5. Top_losses de la segunda clasificación del bloque 2.	64
Figura 6.6. Show_results del bloque 2.	65
Figura 6.7. Generación de pares de imágenes.	67
Figura 6.8. Disposición espacial de las dos primeras componentes principales de las imágenes del bloque 3.	69
Figura 6.9. Disposición espacial de las dos primeras componentes principales de las imágenes del bloque 4.	73
Figura 6.10. Disposición espacial de las dos primeras componentes principales de las imágenes del bloque 5.	78
Figura 6.11. Disposición espacial de las dos primeras componentes principales de las imágenes del bloque 6.	82

Figura 6.12. Comparativa de las distancias normalizadas (D_N) de los bloques 3, 4, 5 y 6 para las cinco clases de linfomas. _____ 86

Figura 6.13. Comparativa de las distancias (D) de las respectivas clases de los bloques 3, 4, 5 y 6. 87

Índice de tablas

Tabla 3.1. Porcentajes normales de leucocitos en sangre de un ser humano adulto (3).	8
Tabla 5.1. Listado de imágenes de leucocitos presentes en pacientes con las patologías que se estudian en este proyecto.	40
Tabla 5.2. Listado de imágenes de células linfoides anormales asociadas a las patologías.	40
Tabla 6.1. Entrenamiento del modelo del bloque 1 utilizando fit_one_cycle durante 4 epoch.	55
Tabla 6.2. Matriz de confusión normalizada de la clasificación inicial del bloque 1.	56
Tabla 6.3. Entrenamiento del modelo del bloque 1 utilizando fit_one_cycle durante 1 epoch después del unfreeze.	57
Tabla 6.4. Entrenamiento del modelo del bloque 1 utilizando fit_one_cycle durante 2 epoch después de encontrar el learning rate.	58
Tabla 6.5. Matriz de confusión normalizada de la segunda clasificación del bloque 1.	58
Tabla 6.6. Entrenamiento del modelo del bloque 2 utilizando fit_one_cycle durante 4 epoch.	61
Tabla 6.7. Matriz de confusión normalizada de la clasificación inicial del bloque 2.	61
Tabla 6.8. Entrenamiento del modelo del bloque 2 utilizando fit_one_cycle durante 1 epoch después del unfreeze.	62
Tabla 6.9. Entrenamiento del modelo del bloque 2 utilizando fit_one_cycle durante 2 epoch después de encontrar el learning rate.	63
Tabla 6.10. Matriz de confusión normalizada de la segunda clasificación del bloque 2.	63
Tabla 6.11. Comparativa de los resultados de los bloques 1 y 2.	66
Tabla 6.12. Entrenamiento del modelo del bloque 3 utilizando fit_one_cycle durante 5 epoch.	67

Tabla 6.13. Entrenamiento del modelo del bloque 3 utilizando fit_one_cycle durante 1 epoch después del unfreeze. _____	68
Tabla 6.14. Entrenamiento del modelo del bloque 3 utilizando fit_one_cycle durante 5 epoch después de encontrar el learning rate. _____	68
Tabla 6.15. Resultados función oneshot para la clase Burkitt (Bloque 3). _____	69
Tabla 6.16. Resultados función oneshot para la clase Folicular (Bloque 3). _____	70
Tabla 6.17. Resultados función oneshot para la clase HCL (Bloque 3). _____	70
Tabla 6.18. Resultados función oneshot para la clase LCM (Bloque 3). _____	71
Tabla 6.19. Resultados función oneshot para la clase LLC (Bloque 3). _____	71
Tabla 6.20. Entrenamiento del modelo del bloque 4 utilizando fit_one_cycle durante 5 epoch. _____	72
Tabla 6.21. Entrenamiento del modelo del bloque 4 utilizando fit_one_cycle durante 1 epoch después del unfreeze. _____	72
Tabla 6.22. Entrenamiento del modelo del bloque 4 utilizando fit_one_cycle durante 5 epoch después de encontrar el learning rate. _____	73
Tabla 6.23. Resultados función oneshot para la clase Burkitt (Bloque 4). _____	74
Tabla 6.24. Resultados función oneshot para la clase Folicular (Bloque 4). _____	74
Tabla 6.25. Resultados función oneshot para la clase HCL (Bloque 4). _____	75
Tabla 6.26. Resultados función oneshot para la clase LCM (Bloque 4). _____	75
Tabla 6.27. Resultados función oneshot para la clase LLC (Bloque 4). _____	76
Tabla 6.28. Entrenamiento del modelo del bloque 5 utilizando fit_one_cycle durante 2 epoch. _____	76

Tabla 6.29. Entrenamiento del modelo del bloque 5 utilizando fit_one_cycle durante 4 epoch después del unfreeze. _____	77
Tabla 6.30. Entrenamiento del modelo del bloque 5 utilizando fit_one_cycle durante 4 epoch después de encontrar el learning rate. _____	77
Tabla 6.31. Resultados función oneshot para la clase Burkitt (Bloque 5). _____	78
Tabla 6.32. Resultados función oneshot para la clase Folicular (Bloque 5). _____	79
Tabla 6.33. Resultados función oneshot para la clase HCL (Bloque 5). _____	79
Tabla 6.34. Resultados función oneshot para la clase LCM (Bloque 5). _____	80
Tabla 6.35. Resultados función oneshot para la clase LLC (Bloque 5). _____	80
Tabla 6.36. Entrenamiento del modelo del bloque 6 utilizando fit_one_cycle durante 2 epoch. _	81
Tabla 6.37. Entrenamiento del modelo del bloque 6 utilizando fit_one_cycle durante 4 epoch después del unfreeze. _____	81
Tabla 6.38. Entrenamiento del modelo del bloque 6 utilizando fit_one_cycle durante 4 epoch después de encontrar el learning rate. _____	82
Tabla 6.39. Resultados función oneshot para la clase Burkitt (Bloque 6). _____	83
Tabla 6.40. Resultados función oneshot para la clase Folicular (Bloque 6). _____	83
Tabla 6.41. Resultados función oneshot para la clase HCL (Bloque 6). _____	84
Tabla 6.42. Resultados función oneshot para la clase LCM (Bloque 6). _____	84
Tabla 6.43. Resultados función oneshot para la clase LLC (Bloque 6). _____	85
Tabla 6.44. Comparativa de los resultados obtenidos con la función oneshot de los bloques 3, 4, 5 y 6. _____	86

Tabla 9.1. Costes directos de hardware.	93
Tabla 9.2. Costes directos de personal.	94
Tabla 9.3. Costes indirectos (101)(102).	94
Tabla 9.4. Presupuesto total.	95

1. Prefacio

1.1. Origen del trabajo

Este trabajo surge de la necesidad de agilizar el diagnóstico de linfomas con una incidencia muy baja creando una metodología automatizada para clasificar células anormales que pueden ayudar a diagnosticar dichos linfomas. Se realiza en colaboración con el laboratorio Core del Hospital Clínic ya que permitieron el acceso a las distintas imágenes de los leucocitos que han permitido realizar este proyecto.

1.2. Motivación

El cáncer sigue constituyendo una de las principales causas de mortalidad del mundo. El número de tumores continúa creciendo, habiendo aumentado desde los 14 millones de casos en el mundo estimados en el 2012 hasta los 18,1 millones en 2018. Las estimaciones poblacionales indican que el número de casos nuevos aumentará en las dos próximas décadas, alcanzando los 29,5 millones de casos en el 2040 (1). En España el número de tumores diagnosticados en el año 2019 rondará los 280.000.

El diagnóstico del cáncer es un tema con una importancia humana y social muy elevada. A distintos niveles toda la sociedad conoce el impacto y la repercusión que puede tener el cáncer en una persona y su círculo más cercano.

La motivación para llevar a cabo un proyecto de estas características surge al haber realizado diferentes asignaturas del grado como procesamiento de imágenes biomédicas (PIB) y aprendizaje bioestadístico (AB) que permiten ver un punto de vista distinto desde donde los ingenieros pueden aportar su granito de arena a problemas actuales de la medicina que pueden ayudar a mejorar la calidad de vida de las personas.

Este proyecto es una aportación más a la constante lucha contra el cáncer.

1.3. Requisitos previos

Para este proyecto es necesario tener nociones básicas de fisiología que permitan entender las células sanguíneas y la formación de linfomas. Es necesario también tener experiencia con algoritmos de

Machine Learning y *Deep Learning*, entender las aplicaciones del aprendizaje automática y el funcionamiento de redes neuronales que permiten crear los clasificadores de imágenes.

Se requiere haber realizado los cursos online *Practical Deep Learning for Coders* y *Part 2: Deep Learning from the Foundations* ofrecidos por *Fastai*. Es necesario tener experiencia con el lenguaje de programación Python y estar familiarizado con el uso de librerías como *Pytorch*, *Pandas*, *Numpy*, etc. y el entorno de trabajo *Jupyter Notebook*.

Para realizar el proyecto se ha contado con una estación de trabajo remota. Para ello es necesario familiarizarse con la conexión al servidor *Deepbox* mediante SSH y VPN utilizando el software *F5 Access*.

2. Introducción

2.1. Objetivos del trabajo

Se establece como objetivo principal del proyecto crear un clasificador automático de imágenes de células linfoides anormales en sangre periférica (SP) asociadas a los cinco tipos de linfomas. Para abordar este problema se subdividen los objetivos de la siguiente manera:

1. Utilizar métodos de clasificación de imágenes convencionales para evaluar su desempeño cuando se tiene disponible un conjunto de imágenes muy limitado.
2. Utilizar técnicas de *Few-Shot Learning* (FSL), particularmente redes neuronales siamesas (SNN) para clasificar las células anormales asociadas a los cinco tipos de linfomas, puesto que estas redes neuronales (NN) están diseñadas para trabajar óptimamente con pocos ejemplares de imágenes a clasificar.
3. Comparar los resultados obtenidos para diseñar un clasificador con el mejor rendimiento posible.

2.2. Alcance del trabajo

El alcance del trabajo viene limitado por las características de las imágenes que se tratan. Las imágenes de las células de sangre periférica pertenecen a distintos pacientes del Hospital Clínic y, debido a la ley de protección de datos, se garantiza la privacidad de los pacientes del hospital. Los métodos de clasificación se utilizarán en imágenes cedidas por la Dra. Anna Merino del laboratorio Core del Hospital Clínic. En lo que respecta a las redes neuronales, no se modificará la propia arquitectura de las redes puesto que esto excedería el alcance de un TFG.



3. Bases biológicas

3.1. Sangre y células sanguíneas

La sangre es un tejido compuesto de glóbulos rojos (eritrocitos), glóbulos blancos (leucocitos), plaquetas y otras sustancias suspendidas en un líquido llamado plasma (2), que contiene agua, sales y proteínas.

Las células sanguíneas empiezan a formarse en la médula ósea a partir de un tipo de célula llamado célula precursora hematopoyética pluripotencial, de la cual derivan todas las células de la sangre. La Figura 3.1 muestra las divisiones de las células pluripotenciales para formar las distintas células sanguíneas.

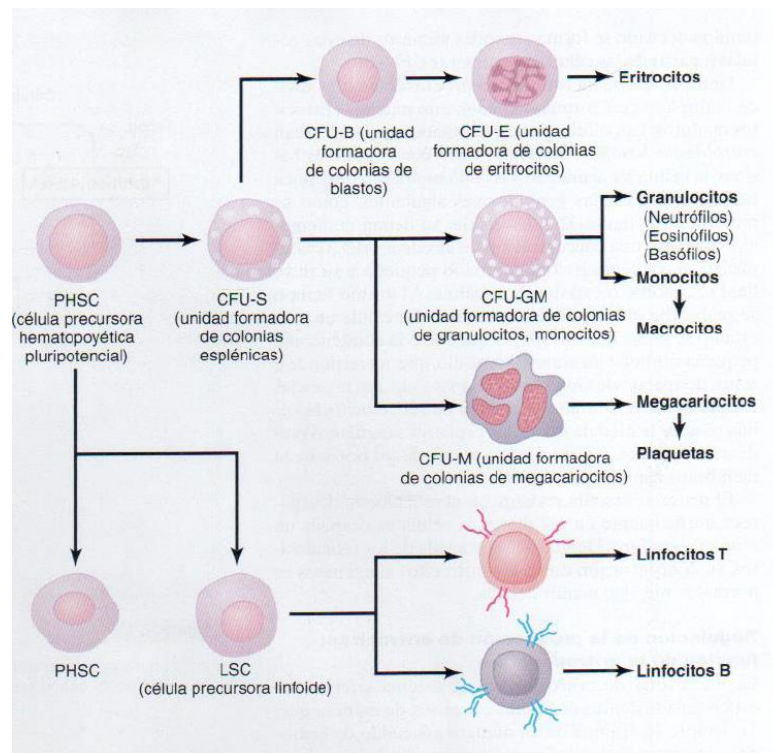


Figura 3.1. Formación de diferentes células sanguíneas a partir de la célula precursora pluripotencial (PHSC) en la médula ósea (3).

A medida que estas células se reproducen, una pequeña parte de ellas permanece igual que las células pluripotenciales originales y se queda en la médula ósea. La gran mayoría de las células se diferencian hasta formar los distintos tipos celulares mostrados en la Figura 3.1.

Las células que se encuentran en un estadio intermedio son muy parecidas morfológicamente a las células precursoras pluripotenciales, aunque ya están comprometidas a una línea celular en particular y por eso reciben el nombre de células precursoras comprometidas (4).

Los inductores del crecimiento son proteínas que controlan el crecimiento y la reproducción de las células precursoras comprometidas. Existen cuatro inductores principales del crecimiento que pueden inducir el crecimiento y reproducción de un tipo específico de célula precursora comprometida. Existen también inductores de crecimiento que favorecen el crecimiento y reproducción de todos los tipos diferentes de células precursoras comprometidas, como la interleucina 3 (5).

Los inductores del crecimiento únicamente favorecen el crecimiento de las células. Los inductores de la diferenciación son los encargados de favorecer la diferenciación de las células.

La formación de estos dos tipos de inductores está controlada por factores externos a la médula ósea. Por ejemplo, en el caso de los eritrocitos, la exposición de la sangre a un nivel bajo de oxígeno durante un largo período de tiempo induce el crecimiento, diferenciación y producción de un número elevado de eritrocitos (6).

3.1.1. Eritrocitos

Los eritrocitos, también conocidos como hematíes, son las células más abundantes de la sangre (constituyen un 40% del volumen sanguíneo) y son necesarias para el transporte de oxígeno a los tejidos (3).

La función principal de los eritrocitos es transportar hemoglobina, proteína que transporta oxígeno desde los pulmones a los tejidos y órganos del cuerpo; además, transporta el dióxido de carbono (CO_2) de vuelta a los pulmones (7).

Los eritrocitos contienen una gran cantidad de anhidrasa carbónica (3), una enzima que cataliza la reacción reversible entre el dióxido de carbono y el agua para formar ácido carbónico (H_2CO_3). La rapidez de esta reacción posibilita que el agua de la sangre transporte grandes cantidades de dióxido de carbono en forma de ion bicarbonato (HCO_3^-) desde los tejidos a los pulmones, donde se convierte en dióxido de carbono y se expulsa a la atmósfera como un producto de desecho del organismo.

3.1.2. Leucocitos

Los leucocitos son una parte importante de la defensa del cuerpo humano contra microorganismos y sustancias extrañas. Los leucocitos se producen principalmente en la médula ósea.

El valor real de los leucocitos es que la mayoría de ellos se transportan específicamente a zonas de infección en inflamación intensas, lo que constituye una defensa rápida y potente frente a los microorganismos infecciosos (3). Los leucocitos, junto a otras células tisulares, evitan enfermedades fagocitando bacterias o virus invasores o bien formando anticuerpos y linfocitos sensibilizados que pueden destruir o inactivar los microorganismos invasores (8).

Existen cinco tipos principales de leucocitos (representados en la Figura 3.2):

- Neutrófilos.
- Eosinófilos.
- Basófilos.
- Monocitos.
- Linfocitos.

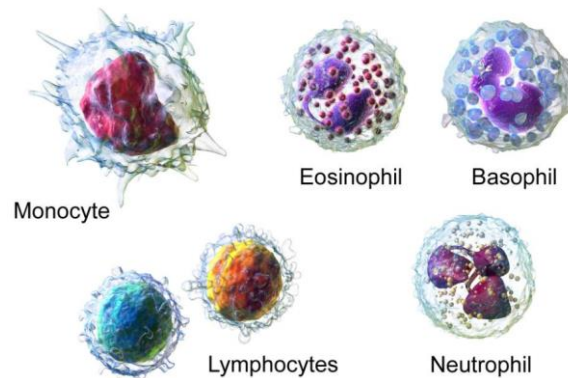


Figura 3.2. Tipos de leucocitos (9).

Los tres primeros tipos de leucocitos (neutrófilos, eosinófilos y basófilos) tienen un aspecto granular por lo que reciben el nombre de granulocitos (ver Figura 3.3).

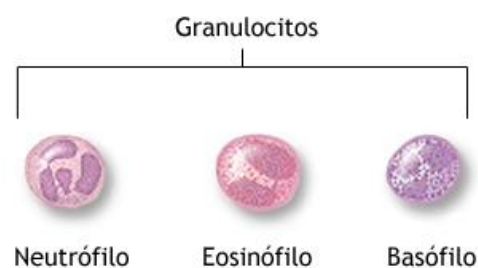


Figura 3.3. Granulocitos (10).

El ser humano adulto tiene unos 7.000 leucocitos por microlitro de sangre (comparado con 5 millones de eritrocitos). Entre todos los leucocitos, los porcentajes normales de los diferentes tipos son

aproximadamente los que se muestran en la Tabla 3.1 (3). El número aproximado de plaquetas por microlitro de sangre es de 300.000 unidades.

Tabla 3.1. Porcentajes normales de leucocitos en sangre de un ser humano adulto (3).

Neutrófilos	62%
Eosinófilos	2,3%
Basófilos	0,4%
Monocitos	5,3%
Linfocitos	30%

Las primeras fases de diferenciación de la célula precursora hematopoyética pluripotencial en los diferentes tipos de células precursoras comprometidas se muestra en la Figura 3.1. Junto a aquellas células comprometidas en la formación de eritrocitos, se forman dos líneas principales de leucocitos, las líneas mielocítica y linfocítica.

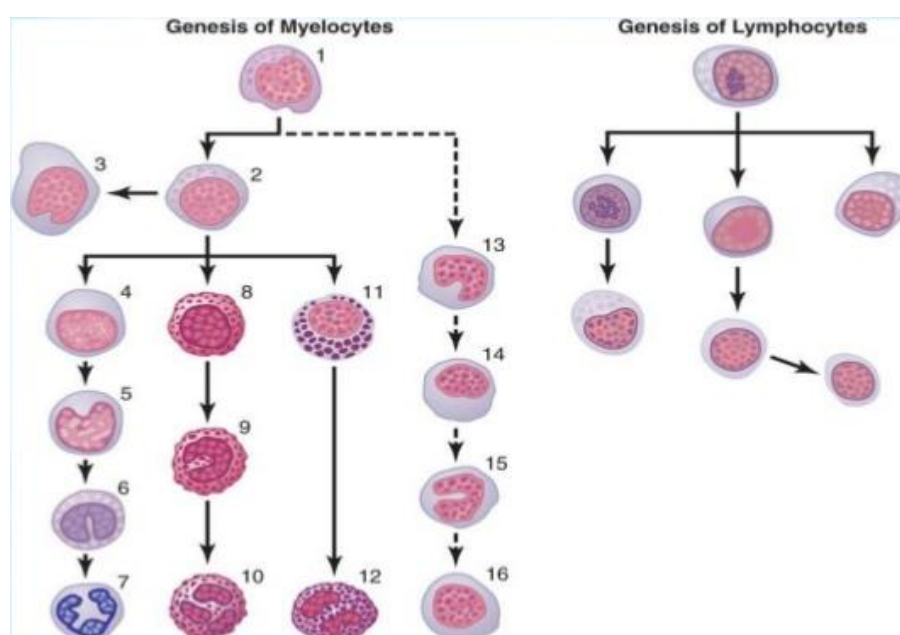


Figura 3.4. Génesis de los leucocitos. Las diferentes células de la serie mielocítica son: 1, el mieloblasto; 2, el promielocito; 3, el megacariocito; 4, el metamielocito neutrófilo; 5, el metamielocito neutrófilo joven; 6, el metamielocito neutrófilo “cayado”; 7, el neutrófilo polimorfonuclear; 8, el mielocito eosinófilo; 10, el eosinófilo polimorfonuclear; 11, el mielocito basófilo; 12, el basófilo polimorfonuclear; 13-16, estadios de formación del monocito (3).

El lado izquierdo de la Figura 3.4 muestra la línea mielocítica, que comienza con el mieloblasto; el lado derecho muestra la línea linfocítica, que comienza con el linfoblasto.

Los granulocitos y los monocitos se forman en la médula ósea; por otro lado, los linfocitos y las células plasmáticas se producen sobre todo en los distintos órganos linfógenos, como los ganglios linfáticos, el bazo, el timo y las amígdalas (ver capítulo 3.2). Los leucocitos formados en la médula ósea se almacenan dentro de la misma hasta que son necesarios en el sistema circulatorio. Los linfocitos se almacenan en su mayoría en varios tejidos linfáticos, excepto un pequeño número que se transporta temporalmente en la sangre (3).

Los linfocitos desempeñan numerosas funciones en el sistema inmunitario, incluyendo la protección contra bacterias, virus, hongos y parásitos.

Existen tres tipos de linfocitos:

- Linfocitos B (células B).
- Linfocitos T (células T).
- Células NK (linfocitos citolíticos naturales).

Aproximadamente el 80% de los linfocitos circulantes son linfocitos T, alrededor del 15% son linfocitos B y el 5% restante son células NK (11).

Para entender la importancia de los linfocitos B y T primeramente se debe comprender el concepto de inmunidad.

La inmunidad es la capacidad del cuerpo humano a resistir los microorganismos y toxinas que tienden a lesionar los tejidos y órganos (3). Existen dos tipos de inmunidad, la inmunidad innata y la inmunidad adquirida.

La inmunidad innata produce una respuesta inmediata ante la presencia de invasores de manera no específica (no necesita que haya habido un encuentro previo con un microorganismo determinado u otro tipo de invasores para producir una respuesta). Participan en esta respuesta distintos tipos de células, como los fagocitos (donde se incluyen los macrófagos, neutrófilos, monocitos y células dendríticas), células NK y sustancias como la histamina (8).

La inmunidad adquirida es específica frente a microorganismos como bacterias, virus y otras toxinas. Existen dos tipos básicos de inmunidad adquirida, la inmunidad humoral y la inmunidad celular.

La inmunidad humoral o inmunidad del linfocito B consiste en la producción de anticuerpos circulantes (moléculas de globulina) en el plasma sanguíneo capaces de atacar al microorganismo invasor. Recibe este nombre porque los linfocitos B son los que producen los anticuerpos.

La inmunidad celular o inmunidad del linfocito T consiste en la formación de un número elevado de linfocitos T activados que se habilitan en los ganglios linfáticos para destruir el microorganismo invasor.

Las principales diferencias entre los linfocitos B y los linfocitos T son las siguientes (12):

- En los linfocitos T, toda la célula desarrolla reactividad frente al antígeno, mientras que los linfocitos B secretan activamente anticuerpos que son las sustancias reactivas.
- Los linfocitos B tienen una diversidad mucho mayor a los linfocitos T, con lo que pueden formar muchos millones de tipos de anticuerpos con diferentes reactividades específicas.

El timo pre-procesa los linfocitos T y el hígado y la médula ósea pre-procesan los linfocitos B.

3.1.3. Plaquetas

Las plaquetas o trombocitos son pequeños fragmentos de células sanguíneas. Las plaquetas se forman en la médula ósea y tienen un papel clave en la coagulación de la sangre. Cuando el número de plaquetas es muy bajo (trombocitopenia), es más probable que aparezcan hematomas en la piel y ocurran sangrados anormales. Cuando el número de plaquetas es muy alto (trombocitemia), la sangre se coagula en exceso, lo que puede desencadenar un accidente isquémico transitorio (13).

3.1.4. Plasma

El plasma es el componente líquido de la sangre donde están suspendidos los eritrocitos, los leucocitos y las plaquetas. Constituye más del 50% de su volumen y está compuesto principalmente por agua, que contiene sales en disolución (electrolitos) y proteínas. La proteína más abundante del plasma es la albúmina, que ayuda a evitar que el líquido se filtre fuera de los vasos sanguíneos y entre en los tejidos, y además cumple funciones de transporte al unirse a sustancias como las hormonas y algunos fármacos (13).

3.2. El sistema linfático

El sistema linfático (ver Figura 3.5) es una red de órganos, ganglios linfáticos, conductos y vasos linfáticos que producen y transportan linfa desde los tejidos hasta el torrente sanguíneo (14). El sistema linfático es una parte esencial del sistema inmunitario del cuerpo humano.

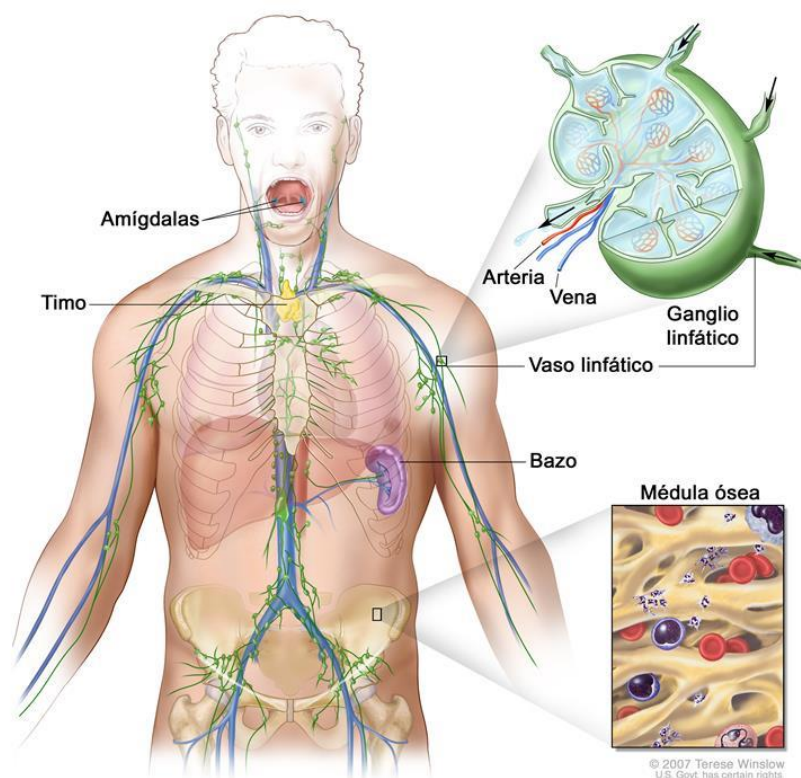


Figura 3.5. Anatomía del sistema linfático (15).

El sistema linfático representa una vía a través de la cual el líquido puede fluir desde los espacios intersticiales hacia la sangre (3).

La linfa es un líquido compuesto por leucocitos (principalmente linfocitos), proteínas, grasas y sales. La linfa circula por los vasos linfáticos aprovechando las contracciones musculares (16). La linfa y los linfocitos se desplazan a través de los vasos linfáticos hasta los ganglios linfáticos, donde los linfocitos destruyen las sustancias dañinas para el organismo.

Los ganglios linfáticos (ver la Figura 3.6) filtran la linfa depurándola de células lesionadas, células cancerosas y partículas extrañas. Los ganglios linfáticos también contienen leucocitos especializados cuya función es englobar y destruir células dañadas, células cancerosas, microorganismos infecciosos y partículas extrañas (17). Los ganglios linfáticos se agrupan en zonas en las que se ramifican los vasos linfáticos, como el cuello, las axilas y las ingles.

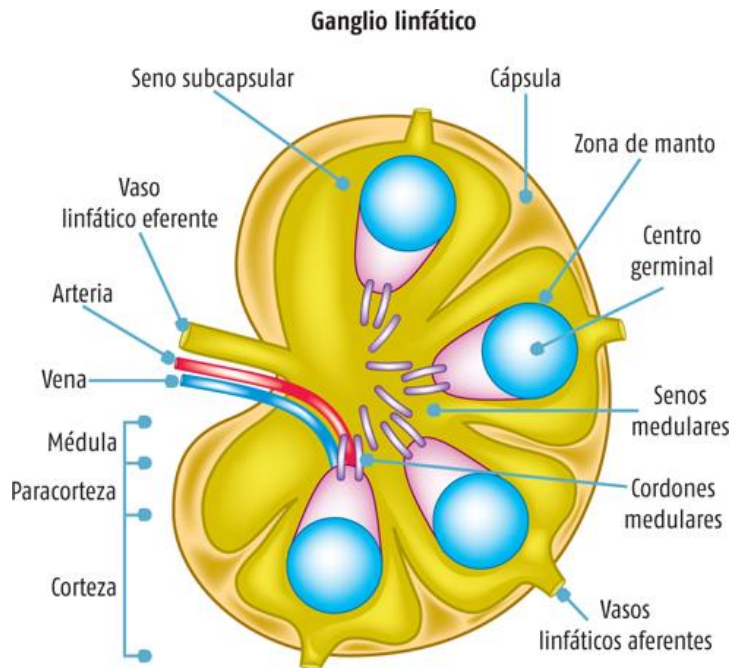


Figura 3.6. Esquema de un ganglio linfático no neoplásico (18).

3.3. Estudio de la sangre

Para la realización de un diagnóstico, una de las primeras pruebas hematológicas a realizar es el frotis sanguíneo, ya que permite evaluar los eritrocitos, los leucocitos y los trombocitos.

El procedimiento para preparar un frotis de sangre es el siguiente (19):

- Extraer una muestra de sangre del paciente.
- Mezclar (sin agitar) la muestra de sangre contenida en el tubo ya que la sangre mal mezclada puede dar lugar a una distribución anormal de las células.
- Colocar una pequeña gota de sangre sobre el extremo de un portaobjetos limpio.
- Utilizar un segundo portaobjetos para distribuir la gota de sangre y extenderla sobre la superficie. Empujar con rapidez el segundo portaobjetos para que la gota de sangre quede “sellada” entre los dos portaobjetos, extendiendo la gota de manera constante y uniforme para formar una película fina de sangre bien distribuida sin agujeros ni surcos.
- Mover el portaobjetos de lado a lado durante varios segundos para secar rápidamente el frotis.

Para poder evaluar el frotis, debe realizarse una tinción del mismo (ver Figura 3.7). Suele utilizarse la tinción de May-Grünwald-Giemsa (MGG) (20) ya que por sus propiedades permite resaltar los componentes básicos de las células sanguíneas.

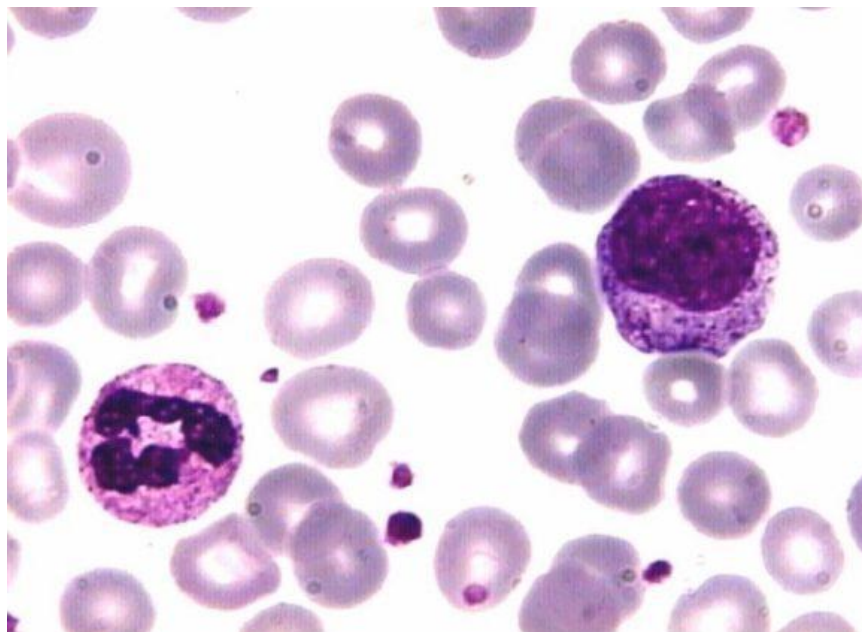


Figura 3.7. Frotis sanguíneo utilizando la tinción de May-Grünwald-Giemsa (20).

El estudio del frotis de sangre permite precisar e informar de las alteraciones morfológicas de las células sanguíneas. Es un procedimiento sencillo, poco costoso y rápido, pero a su vez requiere de mucho cuidado y experiencia debido a la necesidad en la calidad de la extensión y tinción del frotis (21). Además, es esencial que exista una homogeneidad y alta calidad en la preparación de los frotis sanguíneos ya que es el primer paso en la adquisición de las imágenes el cual se explicará en el capítulo 5.1.

3.4. Trastornos de los leucocitos

En este capítulo se introducen los trastornos de los principales tipos de leucocitos (mencionados anteriormente en el capítulo 3.1.2) dependiendo del número de estos en sangre; para posteriormente profundizar en los linfomas.

Normalmente, las personas producen unos 100.000 millones de leucocitos al día (9). Mediante análisis de sangre, puede determinarse la proporción de cada uno de los tipos de leucocitos así como el número total de células de cada tipo en un volumen dado de sangre (suele expresarse en unidades por microlitro de sangre).

Una cantidad anormal de leucocitos (ya sea muy alta o muy baja) indica un trastorno:

- La leucopenia, que es una disminución del número de leucocitos (por debajo de 4.000 células por microlitro de sangre), aumenta la vulnerabilidad de las personas a las infecciones.

- La leucocitosis, que es un aumento del número de leucocitos (por encima de 11.000 células por microlitro de sangre), está causada por una respuesta normal del organismo frente a algunos fármacos o bien para ayudar a combatir una infección.

Sin embargo, también algunas neoplasias de la médula ósea (como la leucemia) provocan un aumento del número de leucocitos en sangre.

3.4.1. Neutrófilos

Los neutrófilos son un tipo de leucocitos que constituyen la principal defensa del organismo contra las infecciones bacterianas agudas y ciertas infecciones por hongos. Ayudan a combatir estas infecciones fagocitando los microorganismos invasores y segregan enzimas que los destruyen (22).

La neutropenia consiste en un número anormalmente bajo de neutrófilos en la sangre. A menudo es un efecto secundario del tratamiento del cáncer con radioterapia o quimioterapia. El límite inferior característico del recuento de neutrófilos es de unas 1.500 células por microlitro de sangre aproximadamente (23). Cuando el recuento de neutrófilos desciende por debajo de esta concentración, el riesgo de infección aumenta. La gravedad de la neutropenia se clasifica de la siguiente manera (24):

- Neutropenia leve (de 1.000 a 1.500 neutrófilos por microlitro de sangre).
- Neutropenia moderada (de 500 a 1.000 neutrófilos por microlitro de sangre).
- Neutropenia grave (por debajo de 500 neutrófilos por microlitro de sangre).

La leucocitosis neutrófila consiste en un número anormalmente alto de neutrófilos en la sangre. La causa más frecuente de un aumento del número de neutrófilos es la respuesta normal del cuerpo a una infección (25).

En la mayoría de los casos, el aumento del número de neutrófilos en sangre es una reacción necesaria del organismo para combatir distintas infecciones. Ahora bien, algunos trastornos inflamatorios como la artritis reumatoide provocan un incremento del número y la actividad de los neutrófilos.

3.4.2. Eosinófilos

Los eosinófilos son un tipo de leucocitos que desempeñan un papel importante en la respuesta del organismo frente a las reacciones alérgicas, el asma y la infección por parásitos (26).

Un número anormalmente bajo de eosinófilos (eosinopenia) en sangre puede observarse en trastornos hormonales como el síndrome de Cushing, en infecciones sanguíneas y en el tratamiento con corticoesteroides. La eosinopenia no suele causar problemas debido a que otras partes del sistema inmunitario compensan adecuadamente esta alteración.

Las causas más comunes de un número elevado de eosinófilos (eosinofilia) son (24):

- Trastornos alérgicos; como el asma, la rinitis alérgica y la dermatitis atópica.
- Infecciones parasitarias.
- Ciertos tipos de cáncer; como el linfoma de Hodgkin, la leucemia y determinados trastornos mieloproliferativos (26).

Si el número de eosinófilos es ligeramente elevado, las personas afectadas no suelen presentar síntomas. Sin embargo, cuando la elevación es muy marcada, el aumento del número de eosinófilos inflama los tejidos y causa daños en órganos, principalmente el corazón, los pulmones y la piel.

3.4.3. Basófilos

Los basófilos son un tipo de leucocitos que tienen algún papel en la protección inmunitaria y la reparación de heridas (27). Los basófilos pueden liberar histamina y otros mediadores, y participan en el inicio de las reacciones alérgicas. La histamina se encarga de aumentar la permeabilidad de los vasos sanguíneos potenciando la respuesta inmune al permitir que las células que actúan en la inflamación alcancen su destino (28).

Un número anormalmente bajo de basófilos (basopenia) puede producirse como respuesta a una tirotoxicosis, infecciones y reacciones de hipersensibilidad aguda.

Un número anormalmente alto de basófilos (basofilia) puede ocurrir en personas con hipotiroidismo (29).

3.4.4. Monocitos

Los monocitos tienen la función de luchar contra determinadas infecciones y ayudan a otros leucocitos a eliminar tejidos muertos o dañados, destruir células cancerosas y regular la inmunidad contra sustancias extrañas (30).

Un número anormalmente bajo de monocitos en sangre (monocitopenia) puede ser debido a cualquier causa de disminución del recuento total de leucocitos (como la neutropenia o la linfocitopenia), como por ejemplo una infección de la sangre, quimioterapia o un trastorno de la médula ósea.

Un número anormalmente alto de monocitos en sangre (monocitosis) puede ser producido como respuesta a infecciones crónicas, enfermedades autoinmunitarias, trastornos de la sangre y determinados cánceres.

3.4.5. Linfocitos

La linfocitopenia consiste en un número anormalmente bajo de linfocitos en sangre. Un número bajo de linfocitos B puede llevar a una disminución en el número de células plasmáticas, por tanto, una reducción en el número de anticuerpos y puede causar un aumento de las infecciones bacterianas. Un número bajo de linfocitos T o de células citolíticas naturales causa problemas para controlar infecciones, en concreto infecciones víricas, parasitarias y fúngicas (31).

La linfocitopenia es más a menudo debida al sida o la desnutrición, aunque también puede ser causada por factores hereditarios, determinadas infecciones, fármacos o trastornos autoinmunitarios (32).

La leucocitosis linfocítica consiste en una cantidad anormalmente alta del número de linfocitos en sangre. La causa más común de un aumento del número de linfocitos son las infecciones víricas como la mononucleosis.

Algunas infecciones bacterianas también provocan un aumento en el número de linfocitos. Ciertos tipos de cáncer, como los linfomas, producen un aumento del número de linfocitos, debido a la liberación de linfoblastos o de células de linfoma hacia el torrente sanguíneo (33).

3.5. Introducción a linfomas

Los linfomas son un tipo de cáncer que puede desarrollarse a partir de los linfocitos B y T.

Existen dos tipos principales de linfoma:

- Linfoma de Hodgkin.
- Linfoma no Hodgkin.

Los síntomas generales que presentan los linfomas son hinchazón de los ganglios linfáticos, pérdida de peso inexplicada, fiebre, sudor nocturno, cansancio, picor y malestar general (34).

El linfoma de Hodgkin se caracteriza por la presencia de unas células cancerosas llamadas células de Reed-Sternberg.

El linfoma de Hodgkin raramente se produce antes de los 10 años de edad y es más frecuente entre los 15 y los 40 años de edad y en las personas mayores de 60 años (35). Generalmente se origina en un ganglio linfático y se extiende de ganglio en ganglio. A medida que la enfermedad avanza, se disemina al hígado, el bazo, la médula ósea y los intestinos (11).

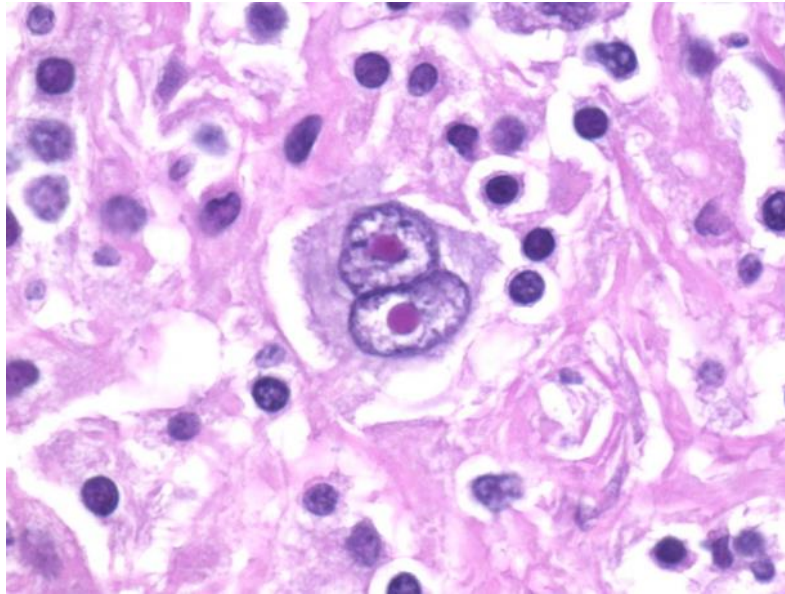


Figura 3.8. Células de Reed-Sternberg (36).

Tal y como se puede ver en la Figura 3.8 la histología muestra grandes linfocitos B modificados con dos núcleos diferentes.

La causa del linfoma de Hodgkin es desconocida, pero la exposición a ciertos virus y bacterias puede desempeñar un papel importante. Las causas infecciosas sospechosas de linfoma de Hodgkin comprenden (35):

- Virus de Epstein Barr.
- *Mycobacterium tuberculosis*.
- Virus del herpes tipo 6.
- Virus de la inmunodeficiencia humana (VIH).

También puede existir una cierta asociación hereditaria. Otras personas con riesgo elevado de sufrir linfoma de Hodgkin son las que tienen (35):

- Síndromes de inmunodeficiencia (tales como ataxia-telangiectasia, síndrome de Klinefelter, síndrome de Chédiak-Higashi y síndrome de Wiskott-Aldrich).
- Trastornos autoinmunitarios tales como artritis reumatoide, celiaquía, síndrome de Sjögren y lupus eritematoso sistémico.
- Trasplante de órganos.
- Exposición previa a fenitoína, radioterapia o quimioterapia.

Los linfomas no hodgkinianos constituyen un grupo diverso de cánceres (más de 50 enfermedades) que se desarrolla en los linfocitos. El cáncer se origina mayoritariamente a partir de los linfocitos B y aproximadamente el 10% de los casos se desarrollan a partir de linfocitos T (37).

A diferencia del linfoma de Hodgkin, el cáncer se origina en el 75% de los casos en los ganglios linfáticos. En un 25% de los casos se desarrolla en otras regiones corporales como el cerebro, los intestinos, el estómago e incluso la glándula tiroidea.

De forma clásica, el linfoma no Hodgkin puede tener un bajo grado de malignidad y tardar años en desarrollarse, o bien tener un alto grado de malignidad y desarrollarse en un período de semanas o meses (38).

A continuación se tratarán los 5 distintos tipos de linfomas que son objeto de estudio del proyecto. Todos ellos son linfomas no hodgkinianos y procedentes de linfocitos B.

3.5.1. Linfoma de Burkitt

El linfoma de Burkitt es un tipo de linfoma agresivo de crecimiento rápido que se presenta en mayor frecuencia en niños y adultos jóvenes. La enfermedad suele afectar la mandíbula, el sistema nervioso central, el intestino, los riñones, los ovarios y otros órganos (39). Existen tres variantes clínicas principales del linfoma de Burkitt (40):

- Linfoma de Burkitt endémico: El linfoma de Burkitt es endémico en África central y representa la malignidad más frecuente entre niños de esa región. Los afectados, por lo general, también tienen una infección crónica de malaria que se cree que es la razón por la que se ha reducido la resistencia al virus de Epstein-Barr. La forma endémica suele manifestarse por agrandamiento de la mandíbula o de los huesos faciales.
- Linfoma de Burkitt esporádico: En el linfoma de Burkitt esporádico predomina la patología abdominal y el tumor puede causar obstrucción intestinal. Pueden estar comprometidos también otros órganos como el cerebro. En adultos, la enfermedad puede ser voluminosa y generalizada, con compromiso masivo del hígado, el bazo y la médula ósea.
- Linfoma de Burkitt asociado a la inmunodeficiencia: En especial por el VIH, por un trasplante o un paciente que esté recibiendo medicamentos inmunodepresores. De hecho, el linfoma de Burkitt tiende a ser la primera manifestación del sida y es frecuente en países occidentales.

En las exploraciones con FDG-PET, los tumores tienen tasas metabólicas elevadas. Se observa una translocación genética característica que involucra el gen *C-myc* del cromosoma 8 y la cadena pesada de inmunoglobulina del cromosoma 14 (40). El linfoma de Burkitt es el tumor humano de crecimiento más rápido, y la anatomía patológica revela un alto índice mitótico, proliferación clonal de linfocitos B

y un patrón en “cielo estrellado” de macrófagos benignos que han englobado linfocitos malignos apoptóticos, tal y como se observa en la Figura 3.9.

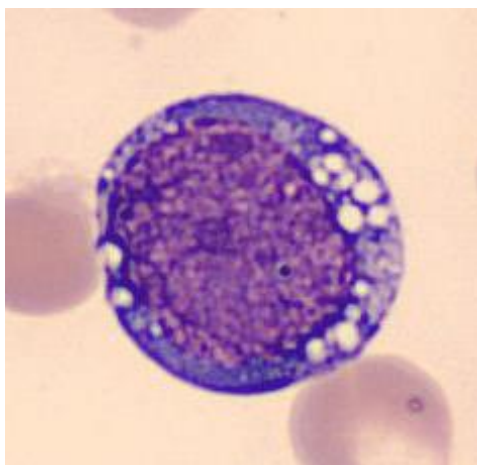


Figura 3.9. Célula linfoide anormal en SP presente en pacientes con linfoma de Burkitt.

3.5.2. Tricoleucemia

La leucemia de células peludas o tricoleucemia (HCL, por sus siglas en inglés: *Hairy Cell Leukemia*) es una enfermedad rara causada por la proliferación anormal de linfocitos B en la médula ósea, en el bazo y en sangre periférica. Es una leucemia linfocítica crónica de progresión muy lenta (41).

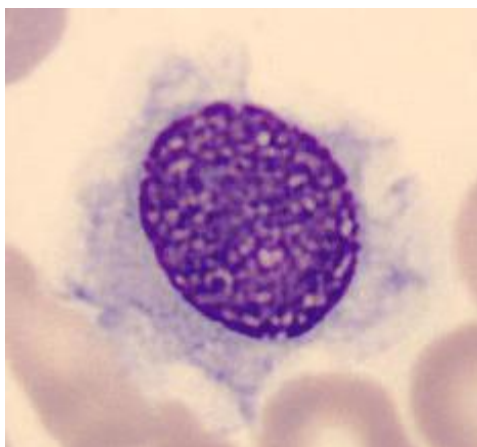


Figura 3.10. Célula linfoide anormal en SP presente en pacientes con tricoleucemia.

El nombre proviene de las proyecciones en forma de cabello que sobresalen de la membrana de estas células (véase Figura 3.10). La HCL representa el 2% de todos los casos de leucemia. Los síntomas de la enfermedad están relacionados con la reducción de la producción normal de células sanguíneas. La baja producción de eritrocitos puede provocar anemia, la baja producción de leucocitos provoca

leucopenia aumentando el riesgo de infecciones, y el bajo número de trombocitos origina hemorragias y una fácil aparición de hematomas (41)(42).

Los síntomas de la HCL incluyen esplenomegalia (agrandamiento patológico del bazo (43)) sin hinchazón de los ganglios linfáticos. El diagnóstico de la HCL puede hacerse mediante una biopsia de la médula ósea. Las células leucémicas muestran un cambio específico en un gen implicado en el crecimiento celular, el gen BRAF (44), que aumenta el crecimiento y la diseminación de las células cancerosas.

3.5.3. Linfoma de células del manto

El linfoma de células del manto (LCM) es el resultado de una transformación maligna de linfocitos B en el borde exterior del folículo de un ganglio linfático llamado zona del manto (ver Figura 3.6) (45). Las células del LCM pueden entrar en los conductos linfáticos y en la sangre, y pueden diseminarse hacia otros ganglios linfáticos o tejidos, como la médula ósea, el hígado y el tubo gastrointestinal.

Alrededor del 85% de los pacientes con LCM presentan una lesión genética característica que afecta el cromosoma 11 y el cromosoma 14 denominada traslocación recíproca t(11;14). Esta traslocación da como resultado el intercambio de segmentos de los cromosomas. En el cromosoma 11 el intercambio ocurre en el lugar del gen de ciclina D1 y en el cromosoma 14 el intercambio ocurre en un gen que controla la formación de moléculas de anticuerpos. La traslocación t(11;14) provoca una superproducción de ciclina D1, proteína que causa la división celular y el crecimiento del tumor (46).

En el 15% restante de los pacientes no se observa la traslocación t(11;14) pero existen otros cambios genéticos que causan la producción excesiva de la ciclina D1.

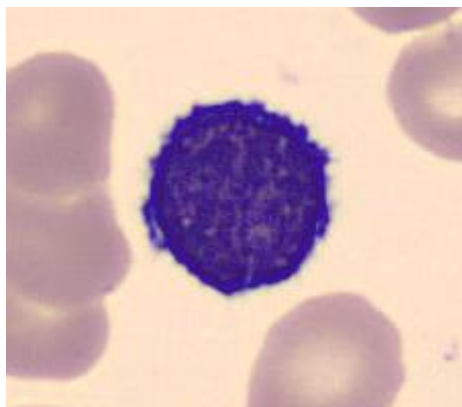


Figura 3.11. Célula linfoide anormal en SP presente en pacientes con linfoma de células del manto.

Este tipo de linfoma se compone de linfocitos tumorales de una morfología característica (ver Figura 3.11), los datos adicionales que dirigen al diagnóstico de LCM es la expresión del marcador CD5, que

puede aparecer en un número reducido de linfomas. En el momento del diagnóstico la mayoría de los pacientes presentan una forma diseminada de la enfermedad. El LCM se asocia frecuentemente con adenopatía generalizada (estado patológico de inflamación de los ganglios linfáticos (47)) y afectación gastrointestinal. Es frecuente también la anemia y la afectación de la médula ósea (48).

El tratamiento del LCM se basa en una quimioterapia intensiva combinada con anticuerpos monoclonales.

3.5.4. Linfoma folicular

El linfoma folicular se caracteriza por una proliferación de células B cuya estructura nodular de la arquitectura folicular se conserva (49). Morfológicamente, el linfoma folicular reemplaza la arquitectura normal del ganglio con una proliferación de folículos neoplásicos (ver Figura 3.12). La prevalencia del linfoma folicular se estima en alrededor de 1 de cada 3.000 casos.

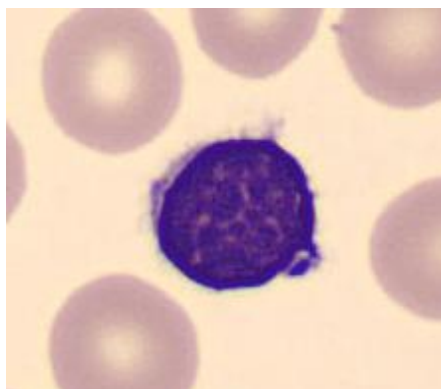


Figura 3.12. Célula linfoide anormal en SP presente en pacientes con linfoma folicular.

El linfoma folicular suele ser indolente y de crecimiento lento (50)(51). Los síntomas aparecen en una fase avanzada de la enfermedad y pueden incluir fiebre, sudores nocturnos y pérdida de peso. En el momento del diagnóstico, generalmente los pacientes aparecen con adenopatía y esplenomegalia.

En el 85% de los casos los linfomas foliculares se asocian con una traslocación $t(14;18)$ que activa el gen BCL2, esencial para algunos procesos de apoptosis o muerte celular programada, causando una sobreexpresión de la proteína BCL2 (52).

A medida que avanza la enfermedad el linfoma folicular puede transformarse en un linfoma difuso de células B grandes (DLBCL), que requiere un tratamiento más agresivo (53).

3.5.5. Leucemia linfocítica crónica

La leucemia linfocítica crónica (LLC) es una enfermedad de progresión lenta, donde los linfocitos maduros se convierten en cancerosos y aumenta su número en sangre, en la médula ósea y en los ganglios linfáticos. Más adelante, estos linfocitos van extendiéndose al hígado y al bazo (54). Los linfocitos cancerosos en la médula ósea sustituyen a las células normales y provocan una disminución en el número de eritrocitos, leucocitos y trombocitos en sangre.

En raras ocasiones, la LLC puede convertirse en un tipo de linfoma de crecimiento rápido. Este fenómeno se conoce como transformación de Richter (55).

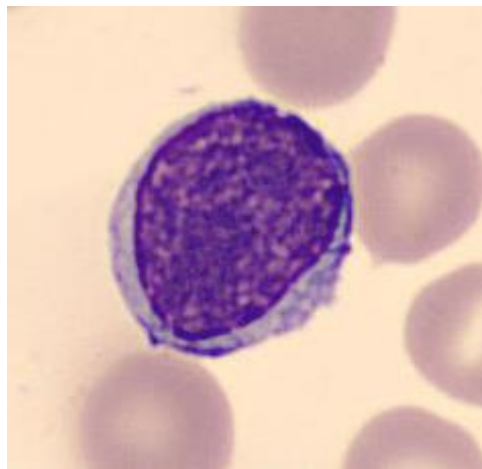


Figura 3.13. Célula linfoide anormal en SP presente en pacientes con leucemia linfocítica crónica.

La LLC es considerada una enfermedad heterogénea puesto que su morfología celular no es siempre uniforme y puede presentar variaciones en sus características (ver Figura 3.13).

En los estadios iniciales de la LLC, la mayoría de las personas no presentan síntomas y la enfermedad se diagnostica solo por el incremento del número de leucocitos. A medida que avanza la enfermedad pueden aparecer los siguientes síntomas (56):

- Hinchazón de los ganglios linfáticos.
- Fatiga.
- Pérdida de peso.
- Dificultad respiratoria.
- Sensación de plenitud abdominal como resultado de un agrandamiento del bazo.

La incidencia de la LLC aumenta con la edad; el 75% de los casos se diagnostica en pacientes mayores de 60 años. Debido a cambios producidos en el sistema inmunitario, las personas con LLC tienen tendencia a desarrollar otros tipos de cánceres, como por ejemplo de piel o de pulmón.

4. Fundamentos de redes neuronales y Few-Shot Learning

Este capítulo sirve de introducción para facilitar la comprensión de las herramientas utilizadas en el diseño del clasificador. Se tratará brevemente el aprendizaje profundo o *deep learning* (DL) y las redes neuronales más adecuadas para tratar problemas de *Few-Shot Learning* (FSL).

El *deep learning* es un tipo de aprendizaje automático o *machine learning* basado en redes neuronales que permite el aprendizaje de un algoritmo o modelo mediante un entrenamiento. El objetivo del modelo es predecir una variable, clasificar una muestra o realizar inferencias sobre los datos (57).

4.1. Introducción a las redes neuronales

Las redes neuronales son sistemas computacionales inspirados en las redes neuronales biológicas presentes en los cerebros de los seres vivos. Estos sistemas “aprenden” a realizar tareas mediante ejemplos, sin la necesidad de ser programados con reglas específicas para estas tareas (58).

4.1.1. Neuronas artificiales

Las unidades fundamentales de las redes neuronales son las neuronas artificiales, también llamadas nodos, interconectados entre sí. Estas neuronas reciben un conjunto de entradas, que pueden venir dadas por la salida de otros nodos o por fuentes externas. Cada una de estas entradas tiene asociado un peso (*weight*), inicializado aleatoriamente, que se va adaptando a medida que se entrena la red neuronal para obtener una mayor precisión (ver Figura 4.1).

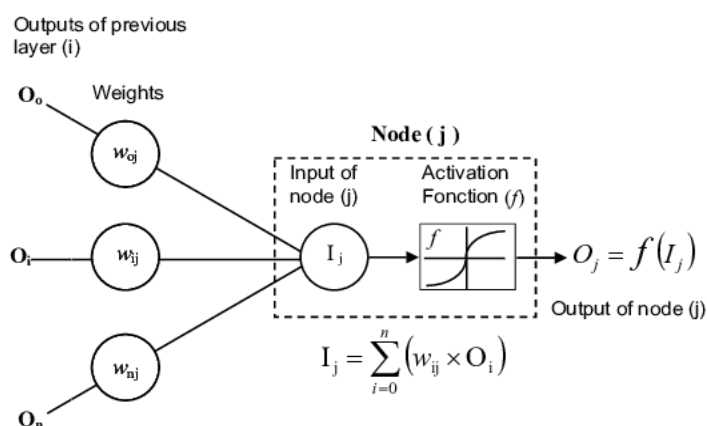


Figura 4.1. Neurona artificial (59).

Este peso indica la importancia que tiene el nodo de donde proviene, a un mayor valor del peso mayor importancia respecto a los otros nodos con valores de peso menores (los pesos pueden tener valores positivos y negativos).

Existe también un parámetro adicional de entrada llamado sesgo (*bias*) que permite ajustar la función de activación del nodo, con lo que se obtiene una mejor función de predicción para la salida de la neurona (60). En la Ecuación 4.1 se muestra el cálculo de la salida de una neurona.

$$Y = \sum (\text{weight} * \text{input}) + \text{bias} \quad (\text{Ec. 4.1})$$

4.1.2. Funciones de activación

Al resultado de la Ecuación 4.1 se le aplica una función de activación. El objetivo de estas funciones de activación es introducir no linealidad a la salida de la neurona; la importancia de esto recae en que la mayoría de datos a tratar no son lineales y se pretende que la red neuronal los pueda representar con la mayor precisión posible. Estas funciones de activación permiten que las diferentes capas de la red neuronal puedan aprender a detectar características y patrones complejos de los datos de entrada.

Existe una función de activación en cada neurona de la red que determina la salida del nodo. Otro aspecto relevante de las funciones de activación es que deben ser eficientes computacionalmente ya que se calculan miles y miles de veces para cada dato de entrada en la red. Las redes neuronales modernas utilizan una técnica llamada *backpropagation* para entrenar el modelo, que permite mejorar la eficiencia de la red neuronal en cada iteración reduciendo el error cometido (61).

Existen diferentes funciones de activación, entre las más utilizadas se encuentran: ReLU, Sigmoide y TanH, ejemplificadas a continuación.

La función de activación ReLU (*Rectified Linear Unit*) (ver Figura 4.2) convierte la entrada a cero si es negativa y la mantiene igual si es positiva. Esta función de activación aumenta en un factor de 6 la rapidez de aprendizaje de la red neuronal respecto a las otras dos funciones de activación estudiadas (62).

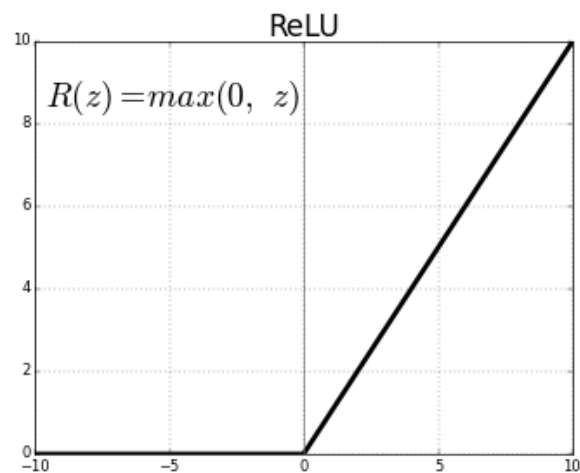


Figura 4.2. Función de activación ReLU (63).

La sigmoide (ver Figura 4.3) se trata de una función de activación no lineal la salida de la cual está contenida en el intervalo $[0, 1]$. Los valores de salida de la función tienden a uno o cero, por lo que es una función que satura y genera gradientes muy pequeños.

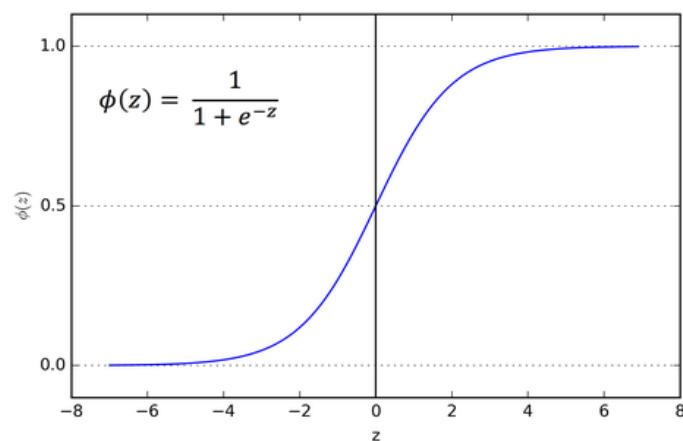


Figura 4.3. Función de activación sigmoide (63).

La tangente hiperbólica (tanH) (ver Figura 4.4) tiene un comportamiento similar a la sigmoide aunque está escalada al intervalo $[-1, 1]$ y genera unos gradientes más elevados.

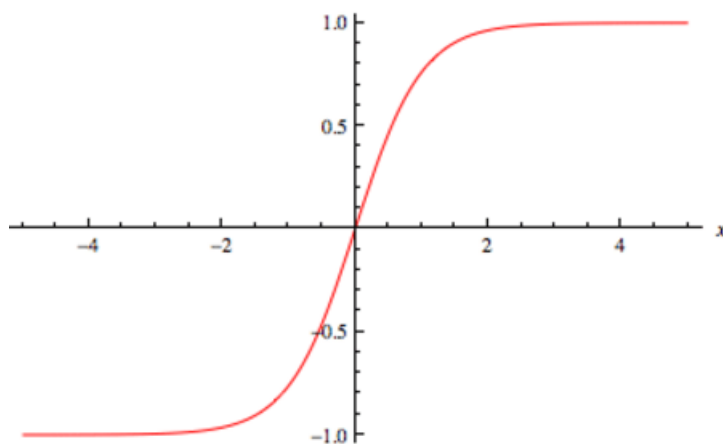


Figura 4.4. Función de activación tanh (63).

4.1.3. Funciones de pérdida

Las funciones de pérdida ayudan a entrenar (ajustar) los parámetros (pesos) de una red neuronal mediante su minimización. Una función de pérdida relaciona el valor real de los datos de entrada de la red con los valores de los datos de salida predichos por la red neuronal (en el caso de este proyecto estos valores hacen referencia a la clase de las imágenes a clasificar) (64). Existen diferentes funciones de pérdida, entre las más utilizadas destacan la *Binary Crossentropy*, *Categorical Crossentropy* y el error cuadrático medio o *Mean Squared Error*.

4.1.4. Métricas, underfitting y overfitting

Las métricas son medidas utilizadas para cuantificar el desempeño o precisión de una red neuronal. Es importante destacar que durante la creación de un modelo existen diferentes subconjuntos de datos a tener en cuenta:

- Conjunto de *training*, utilizado para entrenar el modelo (suele corresponder al 70% del conjunto total de datos aproximadamente).
- Conjunto de *validation*, utilizado para comprobar el desempeño real del modelo y probar parámetros y hiperparámetros de la red neuronal.

Es importante también definir el término *Epoch*. El número de *epoch's* define el total de veces que la red neuronal recorrerá el conjunto de *training*.

La matriz de confusión (65) permite medir el desempeño del modelo computando los valores verdaderos positivos (T_P), verdaderos negativos (T_N), falsos positivos (F_P) y falsos negativos (F_N) (ver Figura 4.5).

		Predicted 0	Predicted 1
Actual 0		TN	FP
Actual 1		FN	TP

Figura 4.5. Ejemplo de matriz de confusión (66).

Para evaluar el desempeño del modelo se utilizará la métrica *accuracy* que permite cuantificar en su totalidad el desempeño del modelo de la siguiente manera:

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \quad (\text{Ec. 4.2})$$

Se utilizará el *error rate* para cuantificar las predicciones incorrectas del modelo:

$$\text{Error rate} = 1 - \text{Accuracy} \quad (\text{Ec. 4.3})$$

La métrica de sensibilidad corresponde al número de predicciones o clasificaciones correctas de una clase con respecto a la totalidad de las clasificaciones de dicha clase. Dicho de otro modo:

$$\text{Sensibilidad}_{\text{Clase } i} = \frac{T_{p_{\text{Clase } i}}}{T_{p_{\text{Clase } i}} + F_{N_{\text{Clase } i}}} \quad (\text{Ec. 4.4})$$

Las métricas ayudan a entender la generalización del modelo mediante su comportamiento y ver si este sufre *underfitting* u *overfitting* (67).

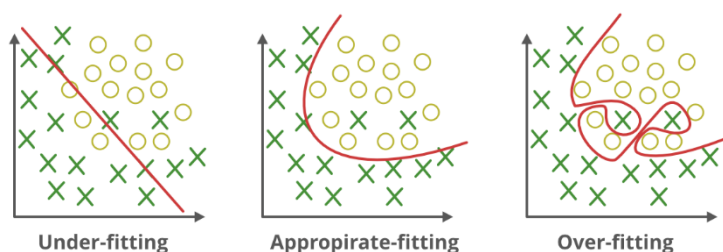


Figura 4.6. Efectos del underfitting y overfitting en un problema de clasificación (68).

En la Figura 4.6 puede observarse como el *overfitting* presenta un ajuste demasiado complejo y el *underfitting* no se ajusta a la realidad de los datos. En la Figura 4.7 puede observarse la evolución del

error de predicción del modelo durante el entrenamiento. Sirve para ilustrar que cuando se produce *overfitting* se obtienen peores resultados debido al sobre-entrenamiento del modelo.

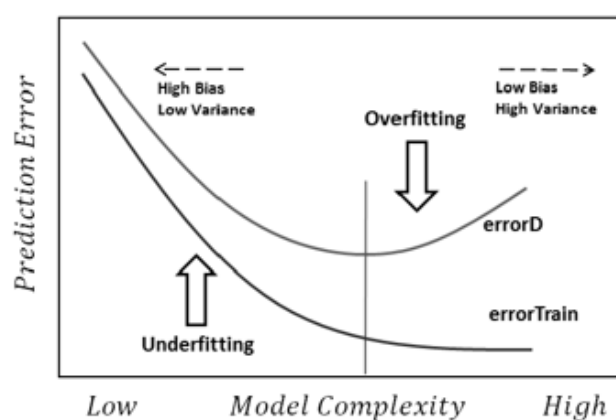


Figura 4.7. Error de predicción de un modelo en función de su complejidad (69).

4.1.5. Tipos de capas

Las redes neuronales están constituidas por un número de capas interconectadas a través de los nodos. Están formadas por tres tipos distintos de capas, tal y como se puede observar en la Figura 4.8:

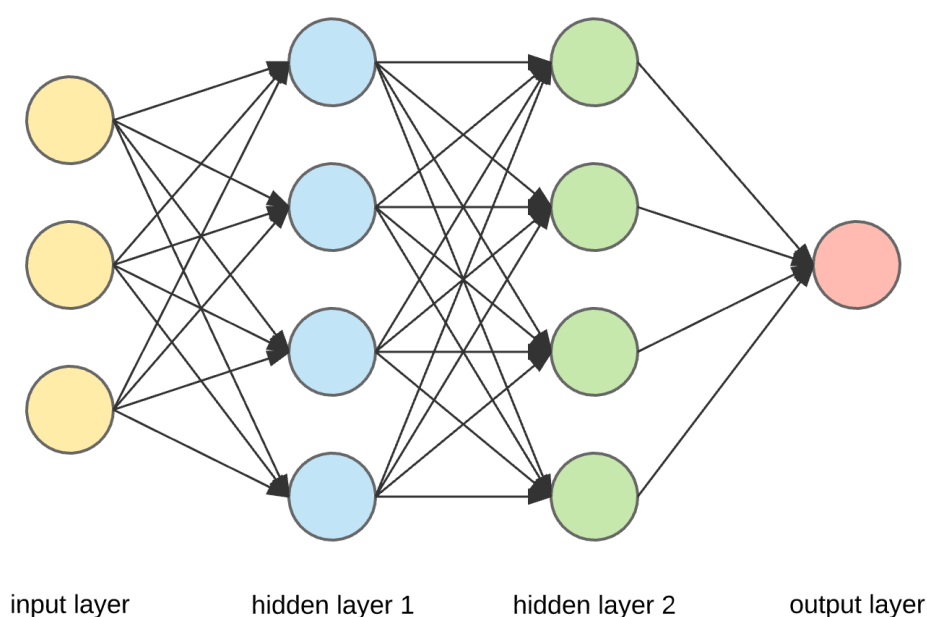


Figura 4.8. Diferentes capas de una red neuronal (58).

- Capa de entrada (*input layer*): Datos iniciales de entrada para la red neuronal.
- Capas ocultas (*hidden layers*): Capas intermedias entre la capa de entrada y la capa de salida.
- Capa de salida (*output layer*): Produce los resultados para los datos de entrada.

La capa de entrada pueden ser imágenes, texto, datos tabulados, etc. Las capas ocultas son las encargadas de llevar a cabo la computación necesaria para el aprendizaje de la red. A diferencia de las capas ocultas, la capa de salida de las redes neuronales no suele tener una función de activación puesto que esta última capa solamente se usa para representar la clase de los datos (en una clasificación, por ejemplo), o algún valor numérico (en el caso de tratarse de un problema de regresión).

En cuanto a la nomenclatura de las redes neuronales, cuando se habla de una red neuronal de N capas, no se tiene en cuenta la capa de entrada, por lo que la red neuronal de la Figura 4.8 tendría tres capas.

Existen muchos tipos de combinaciones entre las diferentes capas de una red neuronal. Estas combinaciones entre las neuronas permiten aproximar funciones que de otra manera serían demasiado complejas. A continuación se explicarán las distintas capas, en función de estas conexiones, que tienen más relevancia para este proyecto:

- En las capas *fully-connected* (o *densamente conectadas*) cada neurona está conectada a todas las neuronas de la capa anterior (es el caso de la red de la Figura 4.8) (70).
- Las capas convolucionales están presentes en las redes neuronales convolucionales (CNN), utilizadas principalmente en problemas de clasificación de imágenes. A diferencia de las capas *fully-connected*, las neuronas de estas capas solo están conectadas a una región de la siguiente capa, dando como salida un mapa de características o imagen convolucionada. Estas convoluciones consisten en pasar una máscara o filtro (también llamado *kernel*) por la imagen realizando productos matriciales; la salida de los cuales originan los píxeles de las imágenes resultantes de la operación. Cabe destacar un parámetro llamado *stride* que indica los saltos de píxeles que da el filtro por la imagen de entrada en cada multiplicación (71).
- Las capas de *pooling* o agrupación se insertan entre capas de convolución consecutivas. La función de estas capas es submuestrear, es decir, reducir el tamaño de la imagen de entrada para reducir la cantidad de parámetros de la red y reducir la carga computacional (72).

4.2. ResNet

Una red neuronal convolucional (CNN) es un tipo de red neuronal artificial cuya aplicación es realizada en matrices bidimensionales, por lo que es muy efectiva para realizar tareas de visión artificial, como por ejemplo la clasificación y segmentación de imágenes.

Tal y como se ha comentado en el capítulo 4.1.2, uno de los métodos más utilizados en redes neuronales es la retropropagación o *backpropagation* para reducir el error cometido por la red neuronal. Ahora bien, cuando una red es muy profunda, los gradientes desde donde se calcula la función de pérdida tienden a cero rápidamente después de varias aplicaciones de la regla de la cadena.

Esto provoca que las neuronas de las primeras capas aprendan a un ritmo mucho menor a las neuronas de las últimas capas; lo que supone un enorme problema, ya que las primeras capas de una red neuronal son las más lentas de entrenar. Estas primeras capas de neuronas son importantes ya que son las responsables de aprender y detectar patrones simples y fundamentan toda la red neuronal.

Todo esto implica que el proceso de aprendizaje de la red neuronal se demora demasiado y el modelo pierde precisión (ver Figura 4.9), es conocido como *The Vanishing Gradient Problem* (73).

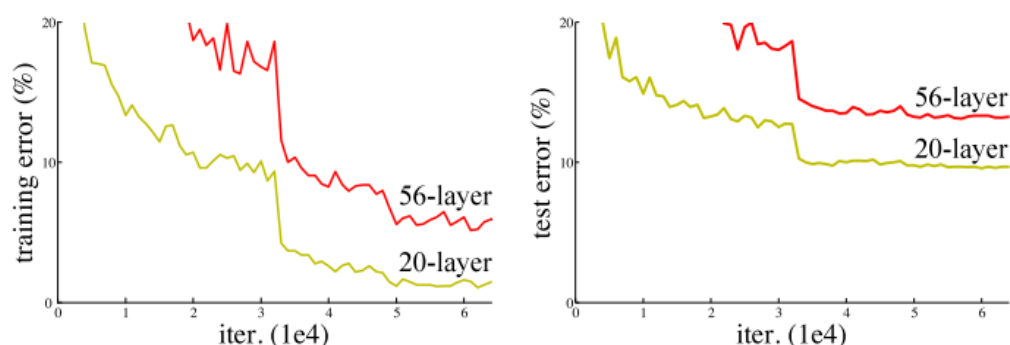


Figura 4.9. Error del modelo frente a la profundidad de la red neuronal (74).

Las arquitecturas ResNet (son un tipo de CNN) permiten utilizar bloques residuales o “*identity shortcut connections*”, posibilitando que los gradientes “salten” a través de varias capas utilizando estas conexiones (ver Figura 4.10) (75).

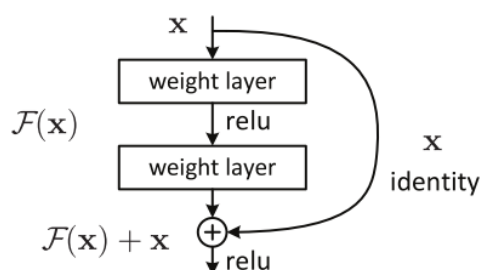


Figura 4.10. Bloque residual (75).

En la Figura 4.11 puede observarse la arquitectura de una red neuronal de 34 capas (izquierda) y la arquitectura de una ResNet34 (derecha).

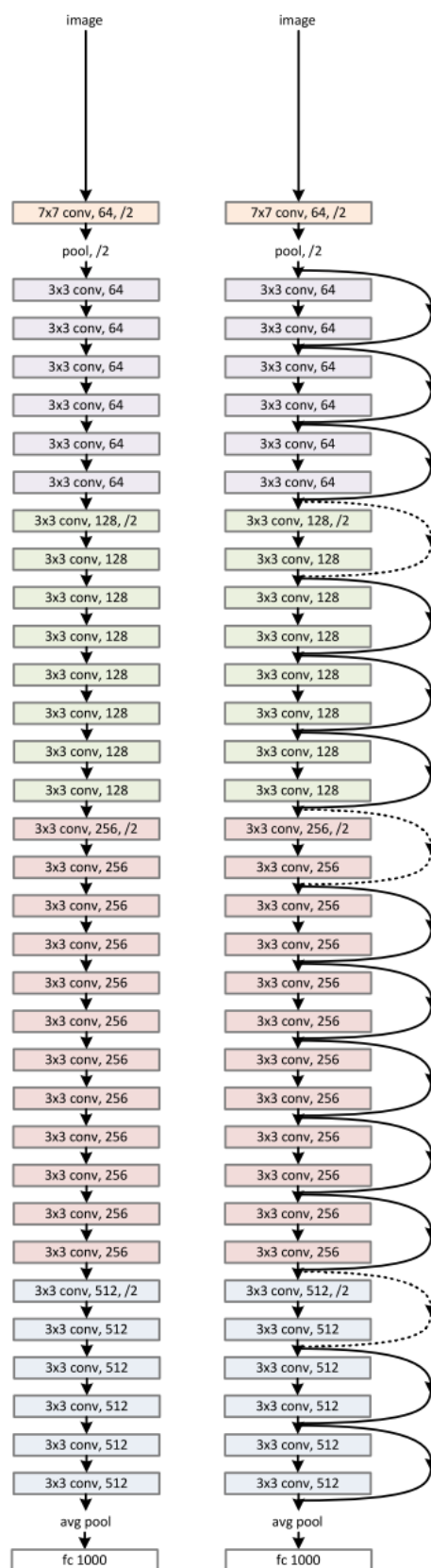


Figura 4.11. Arquitecturas de una red neuronal de 34 capas normal (izquierda) y una ResNet34 (derecha) (75).

Puesto que las ResNet pueden ser de distintos tamaños dependiendo de su número de capas, se utilizará la ResNet34 para explicar su arquitectura (en este proyecto se utilizarán la ResNet34 y la ResNet50).

Se observa en la Figura 4.11 que la ResNet34 consiste en un paso de convolución y *pooling* seguido de cuatro capas de comportamiento similar. Cada una de las capas sigue el mismo patrón; ejecutan una convolución 3x3 con un mapa de características fijo de dimensión F [64, 128, 256, 512] respectivamente, saltándose el *input* cada dos convoluciones. Las dimensiones de anchura y altura se mantienen constantes a lo largo de la capa (76).

La línea discontinua de la Figura 4.11 indica un cambio de dimensión del volumen de entrada (se trata de una reducción del volumen ya que se trata de una convolución). Cabe destacar que esta reducción de volumen se consigue incrementando el *stride* del filtro de uno a dos en la convolución inicial de cada una de las capas, y no mediante una agrupación o *pooling*.

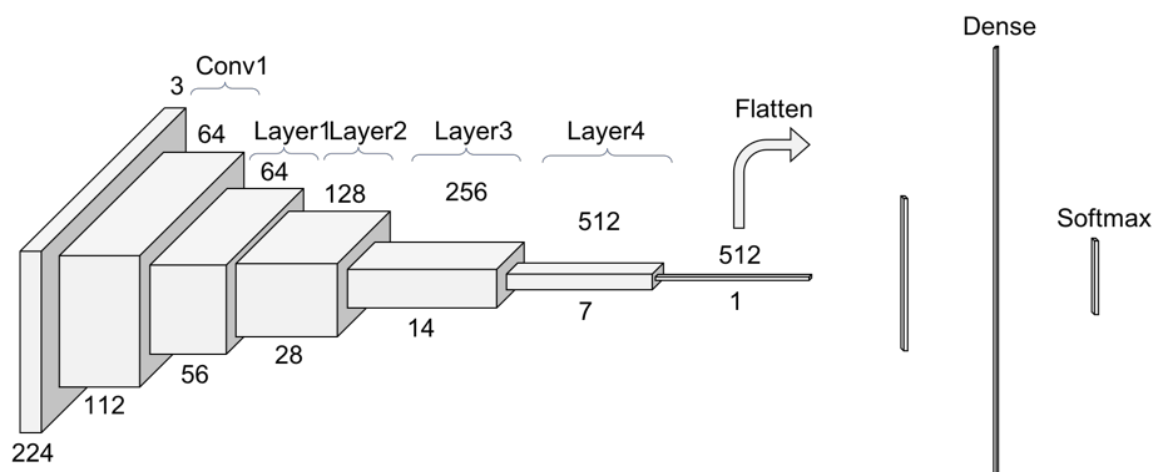


Figura 4.12. Visualización de la ResNet34 (76).

En la Figura 4.12 puede observarse de una manera más entendible el funcionamiento de un modelo convolucional. En el caso de que las imágenes de entrada de la red no sean de 224x224 píxeles habría que modificar la estructura de la ResNet.

4.2.1. Funcionamiento de la ResNet34

El primer paso en la ResNet antes de entrar a las capas (layer) que tienen un comportamiento común es un bloque (llamado *Conv1* en la Figura 4.12) que consiste en una convolución, una normalización del lote o *batch* y una operación de *pooling*.

Primeramente, hay una operación de convolución. Puede observarse en la Figura 4.11 que se ha utilizado un *kernel* de tamaño 7 y un mapa de características de tamaño 64. Se realiza un relleno o

padding tres veces en cada dimensión. Teniendo esto en cuenta, puede observarse en la Figura 4.13 que la salida de la operación será un volumen de 112x112x64.

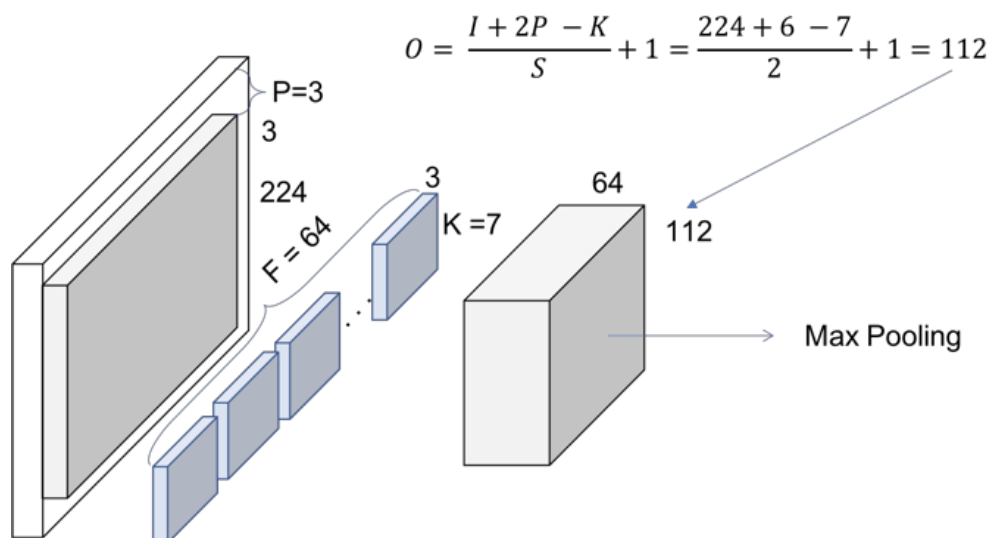


Figura 4.13. Conv1 – Convolución (76).

El siguiente paso es la normalización del *batch*, que al tratarse de una operación elemento a elemento no cambia las dimensiones del volumen. Finalmente hay una operación de *pooling* (3x3) con un *stride* de 2. Al haber realizado un *padding* al volumen de entrada, el volumen de salida tiene las dimensiones deseadas, tal y como se puede observar en la Figura 4.14.

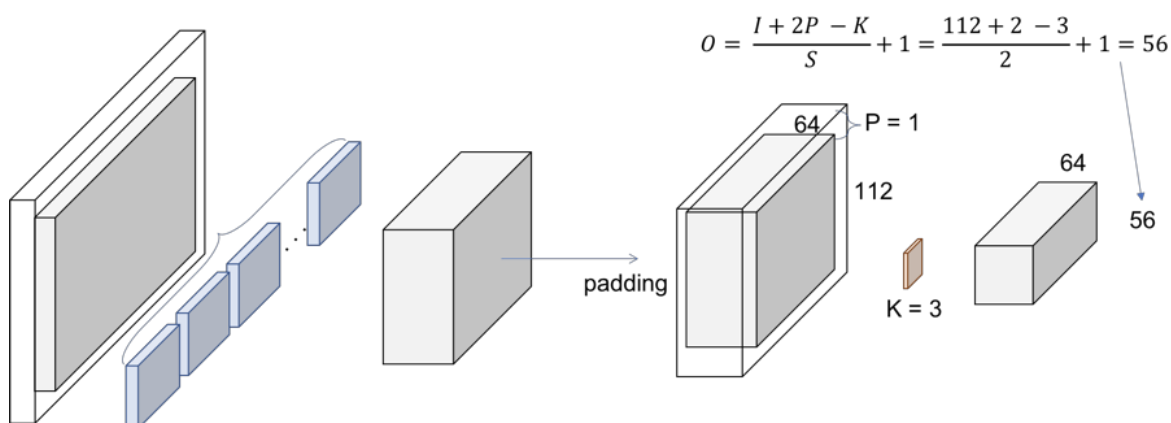


Figura 4.14. Conv1 – Pooling (76).

A continuación se encuentran los bloques llamados *Layer1*, *Layer2*, *Layer3* y *Layer4* (ver Figura 4.12). Cada uno de estas capas de la ResNet está compuestas por diversos bloques. En el caso de utilizar una ResNet más profunda no necesariamente se aumenta el número de capas de la red; sino que aumenta el número de operaciones dentro de cada bloque.

Se entiende por operación el conjunto de una convolución, una normalización del *batch* y una función de activación ReLU a una entrada.

El volumen de entrada para la siguiente operación es el volumen de salida de *Conv1* (ver Figura 4.14, Figura 4.15).

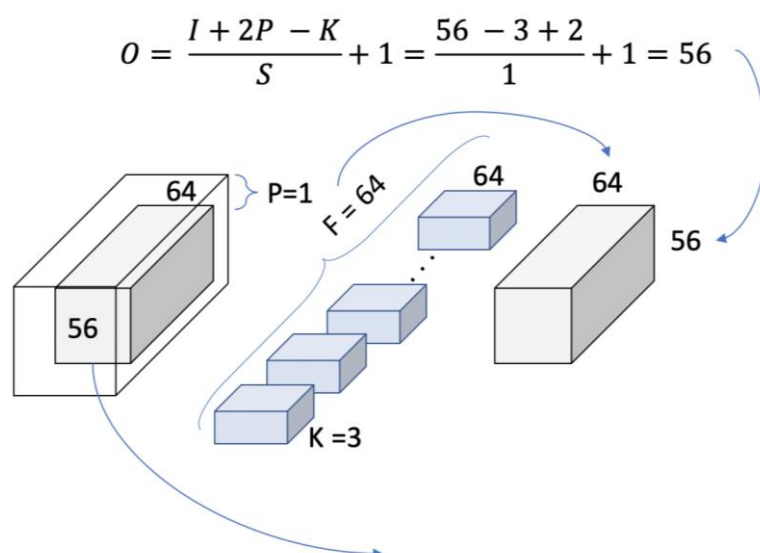


Figura 4.15. Layer1, bloque 1, operación 1 (76).

Se observa en la Figura 4.15 que se utiliza un *kernel* de [3x3, 64], que las dimensiones de salida son [56x56] y que las dimensiones del volumen no varían dentro de un mismo bloque (ya que se utiliza un *padding* igual a 1 y un *stride* de 1).

En la Figura 4.16 se observa cómo se traslada esta operación al bloque 1 entero.

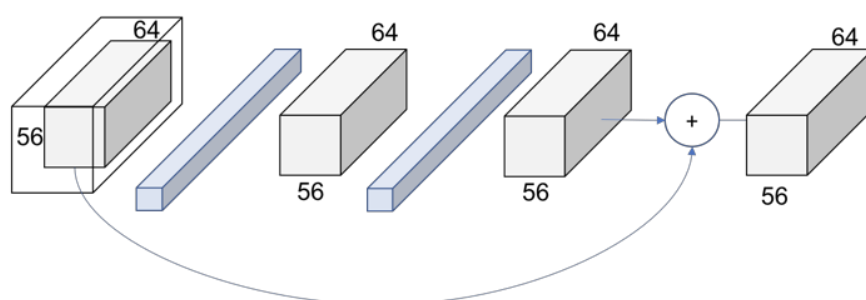


Figura 4.16. Layer1, bloque 1 (76).

De la misma manera, en la Figura 4.17, se visualiza en su totalidad la *Layer1*.

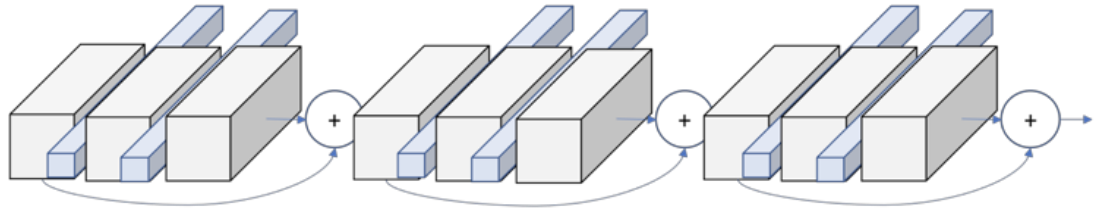


Figura 4.17. Layer 1 (76).

El siguiente paso consiste en escalar del bloque a la capa entera. En la Figura 4.11 puede observarse que las distintas *layer* se diferencian por colores. Se observa también que la primera operación de cada *layer* reduce la dimensión del volumen, por lo que es necesario reescalar también el volumen que pasa por los *Identity Shortcuts* (representados en la Figura 4.11 por las líneas que sobrepasan los bloques por la derecha).

Las líneas discontinuas muestran los *Projection Shortcuts*. Estas conexiones realizan una convolución para asegurar que los volúmenes en esta operación de suma tienen las mismas dimensiones. Existen dos métodos para igualar estas dimensiones; o bien se realiza un *padding* al volumen de entrada o se realizan convoluciones 1x1 (este segundo método puede observarse en la Figura 4.19).

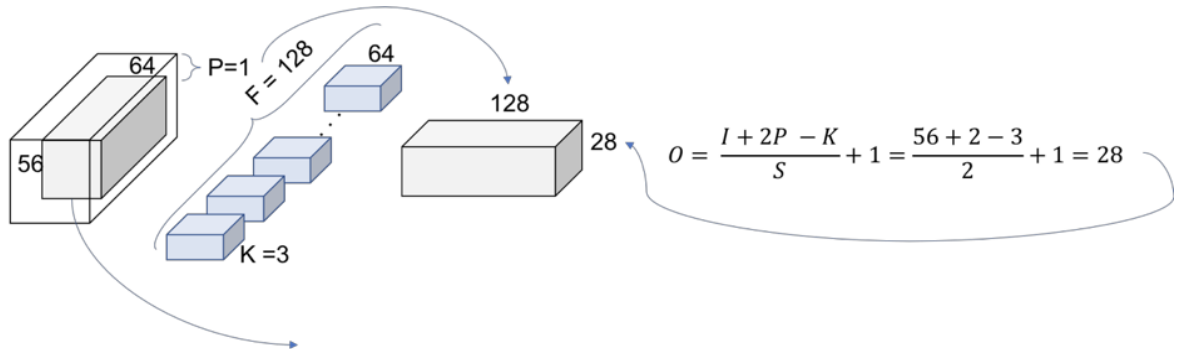


Figura 4.18. Layer 2, bloque 1, operación 1 (76).

La Figura 4.18 muestra esta compresión o down sampling realizada incrementando el *stride* del filtro a 2. La Figura 4.19 muestra la convolución 1x1 realizada para igualar las dimensiones de los volúmenes y poder realizar la operación.

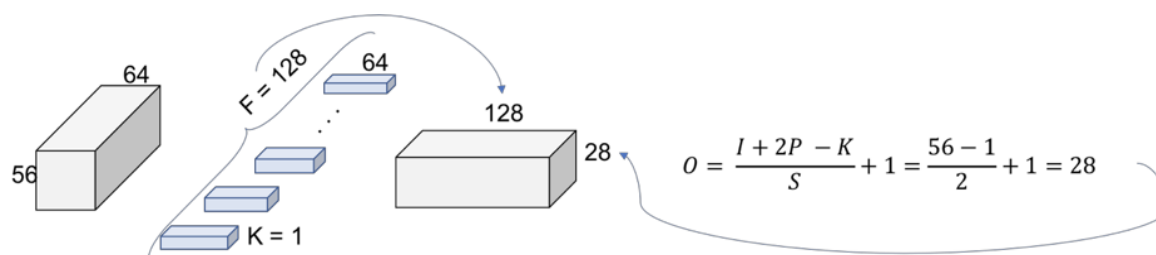


Figura 4.19. Projection Shortcut (76).

La Figura 4.20 muestra como ahora si los dos volúmenes pueden ser sumados al tener las mismas dimensiones.

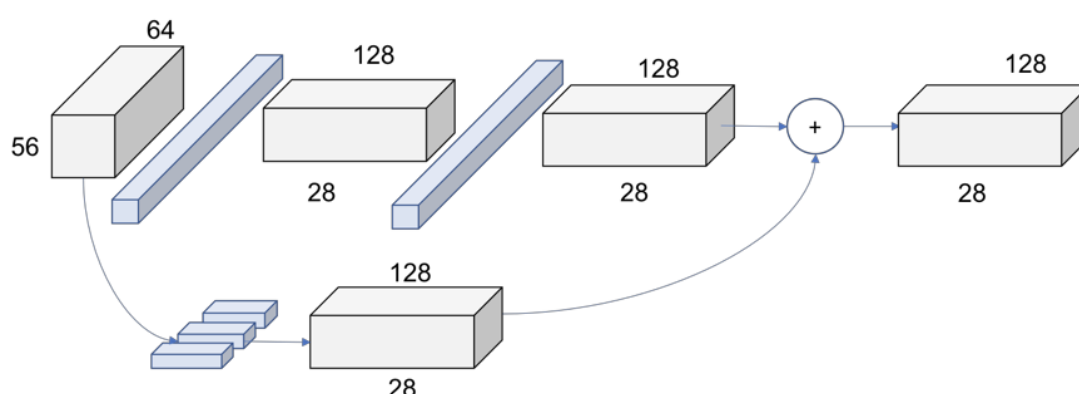


Figura 4.20. Layer 2, bloque 1 (76).

En la Figura 4.21 se observa la totalidad de la *Layer2*.

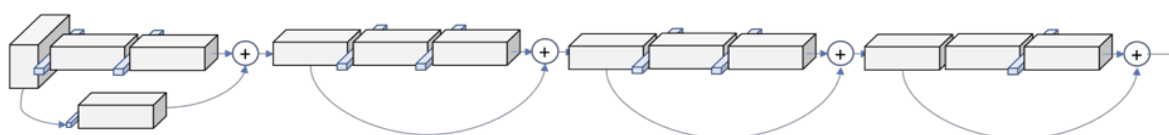


Figura 4.21. Layer 2 (76).

Las siguientes capas (*Layer3* y *Layer4*) funcionan idénticamente, cambiando únicamente las dimensiones de los volúmenes.

4.3. Few-shot learning

Las CNN profundas constituyen hoy en día el estado del arte para resolver problemas de clasificación de imágenes. Ahora bien, una de las limitaciones más grandes es que este tipo de redes requieren un gran número de datos etiquetados (imágenes en este caso). En muchas aplicaciones, reunir tantos datos no es factible, pudiendo ser tan simple como que no existe la cantidad de datos necesarios (como

es el caso de los linfomas con muy baja incidencia). El *Few-Shot Learning* (FSL) tiene como objetivo resolver este problema. El FSL es un método de aprendizaje que consiste en entrenar un modelo con muy pocos datos (77).

4.4. Red Neuronal Siamesa (SNN)

Las redes neuronales siamesas (SNN) fueron utilizadas por primera vez a principios de los años 90 para resolver verificaciones de firmas como un problema de emparejamiento de imágenes (78). Las SNN son redes neuronales que contienen dos o más subredes idénticas. Un ejemplo de la arquitectura de una SNN puede verse a continuación en la Figura 4.22:

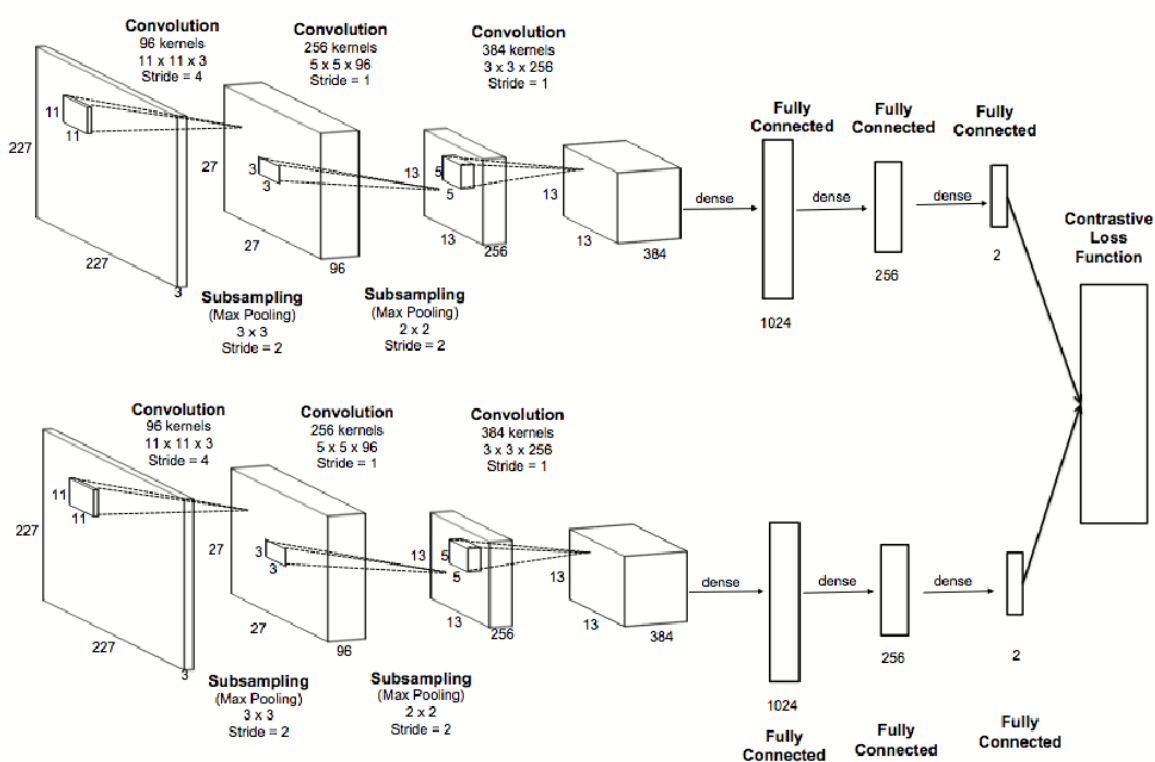


Figura 4.22. Arquitectura de una SNN (79).

Para que una red sea siamesa es necesario que la arquitectura de las subredes que la componen sea idéntica y que también lo sean los pesos (80).

En un problema de clasificación de imágenes normal, la imagen de entrada pasa por el modelo creado y a la salida se genera una función de distribución de probabilidad sobre las distintas clases posibles. Dicho de otro modo, el modelo clasifica la imagen en una clase u otra en función de la probabilidad más elevada de que pertenezca a dicha clase.

En lugar de clasificar directamente una imagen, una SNN utiliza una imagen adicional del conjunto de datos y a la salida de la red se obtiene un índice de similaridad que hace referencia a la probabilidad de que las dos imágenes de entrada pertenezcan a la misma clase. En el capítulo 5 se explica más en profundidad el uso de una SNN para resolver un problema de FSL.

5. Análisis del conjunto de datos y metodología aplicada

5.1. Adquisición de las imágenes

El primer paso en el procesamiento de imágenes consiste en la adquisición de las imágenes. Como se ha comentado en el capítulo 3.3, la preparación de un frotis sanguíneo adecuado y su tinción son pasos esenciales para generar imágenes de calidad.

Es importante también desarrollar una metodología precisa y automatizada para la preparación y tinción de los frotis sanguíneos. El uso de autoanalizadores y preparadores y coloreadores de láminas automatizados (como el Sysmex SP-1000i (81)) proporcionan una forma rápida y consistente de preparar frotis sanguíneos dentro del laboratorio.

Para adquirir imágenes de células sanguíneas el método más utilizado es un microscopio óptico con objetivos de inmersión de 50 a 100 aumentos seguido por una cámara digital con sensores CCD montados en el camino óptico (82). Ahora bien, existen dispositivos digitales de morfología automatizados como el CellaVision DM96 (83) que incorporan un microscopio motorizado con una cámara digital (ver Figura 5.1).

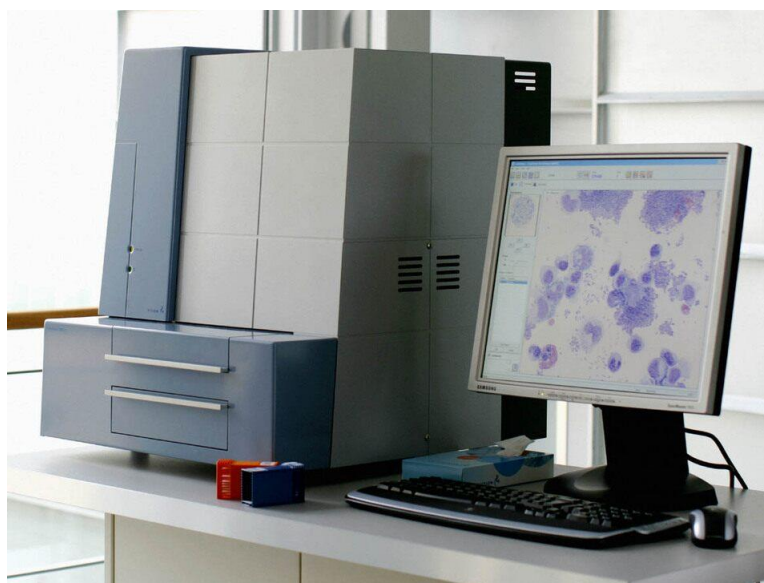


Figura 5.1. CellaVision DM96 (84).

Para este proyecto se han utilizado las imágenes digitales de SP obtenidas en el laboratorio Core del Hospital Clínic utilizando el CellaVision DM96 utilizando un aumento 100x y tinción MGG.

5.2. Análisis del conjunto de datos

El laboratorio Core del Hospital Clínic proporcionó un total de 759 imágenes de 5 clases distintas procedentes de 24 pacientes distintos distribuidas de la siguiente manera:

Tabla 5.1. Listado de imágenes de leucocitos presentes en pacientes con las patologías que se estudian en este proyecto.

Patología	Número de imágenes
Linfoma de Burkitt	188
Linfoma folicular	160
HCL	145
LCM	143
LLC	123

Ahora bien, estas imágenes contienen distintos tipos de leucocitos por lo que ha sido necesario filtrar las imágenes para proceder solamente con las células linfoides; por lo que el nuevo conjunto de datos o *dataset* puede verse reflejado en la Tabla 5.2, donde el total de imágenes es de 501. Las imágenes tienen un tamaño de 363 x 360 píxeles y tienen tres canales (RGB).

Tabla 5.2. Listado de imágenes de células linfoides anormales asociadas a las patologías.

Patología	Número de imágenes
Linfoma de Burkitt	188
Linfoma folicular	149
HCL	48
LCM	59
LLC	57

5.3. Metodología de trabajo

El procedimiento de trabajo se ha dividido en seis bloques principalmente:

- Bloque 1: Afrontar la clasificación de imágenes como un problema de *deep learning* convencional utilizando una ResNet34.
- Bloque 2: Utilizar una ResNet50 para comparar los resultados obtenidos con el primer bloque.
- Bloque 3: Afrontar el problema mediante una SNN con arquitectura ResNet34.
- Bloque 4: Afrontar el problema mediante una SNN con arquitectura ResNet50.
- Bloque 5: Utilizar una SNN con arquitectura ResNet34 pre-entrenada con los pesos obtenidos en el bloque 1.
- Bloque 6: Utilizar una SNN con arquitectura ResNet50 pre-entrenada con los pesos obtenidos en el bloque 2.

Para una mayor visualización de los bloques y el procedimiento se ha creado el siguiente diagrama:

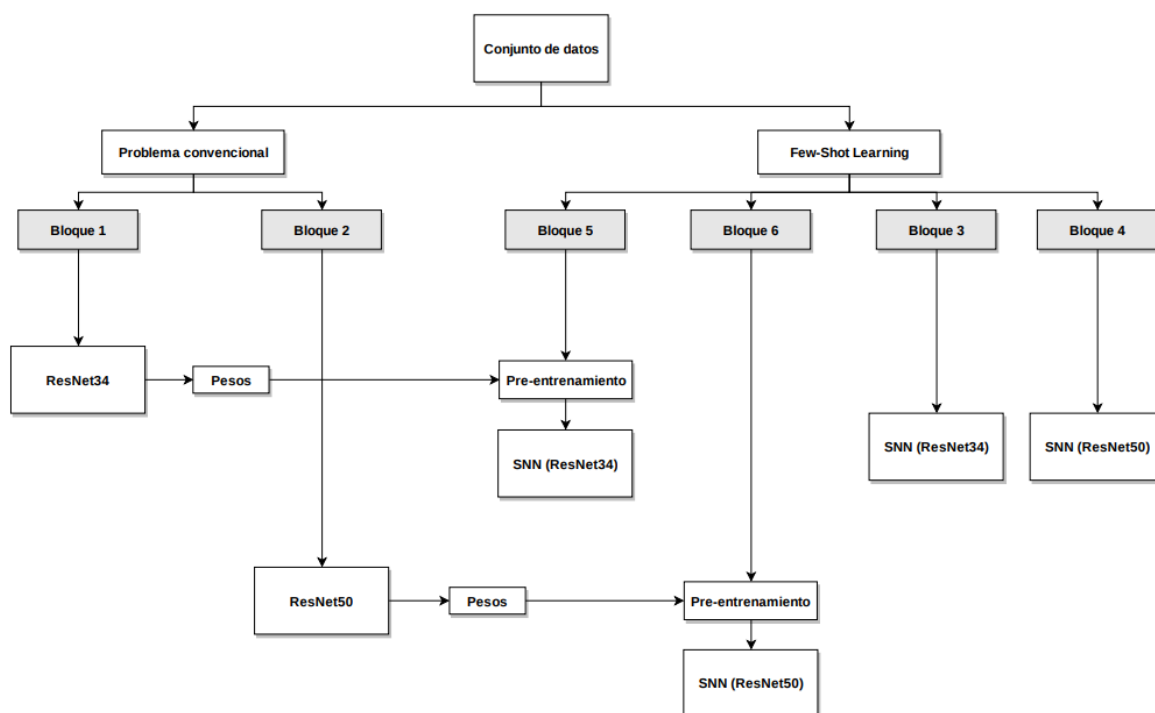


Figura 5.2. Diagrama para visualizar la metodología de trabajo.

El procedimiento desarrollado en esta sección, además de los resultados del capítulo 6, pueden encontrarse en el siguiente repositorio de GitHub:

[Github.com/Marcfont97/TFG](https://github.com/Marcfont97/TFG)

5.3.1. Bloque 1 (ResNet34 convencional)

La principal diferencia encontrada en distintos *datasets* para clasificar imágenes es el modo en el que se almacenan las etiquetas de las clases de las imágenes. En este caso las etiquetas están almacenadas en el propio nombre de las imágenes (por ejemplo, la imagen *BURKITT_32.jpg* corresponde a un linfocito con linfoma de Burkitt).

Para ello, se utiliza la instrucción `pat` de `fast.ai` que nos permite extraer la etiqueta de la clase de la imagen de su propio nombre (ver Figura 5.3).

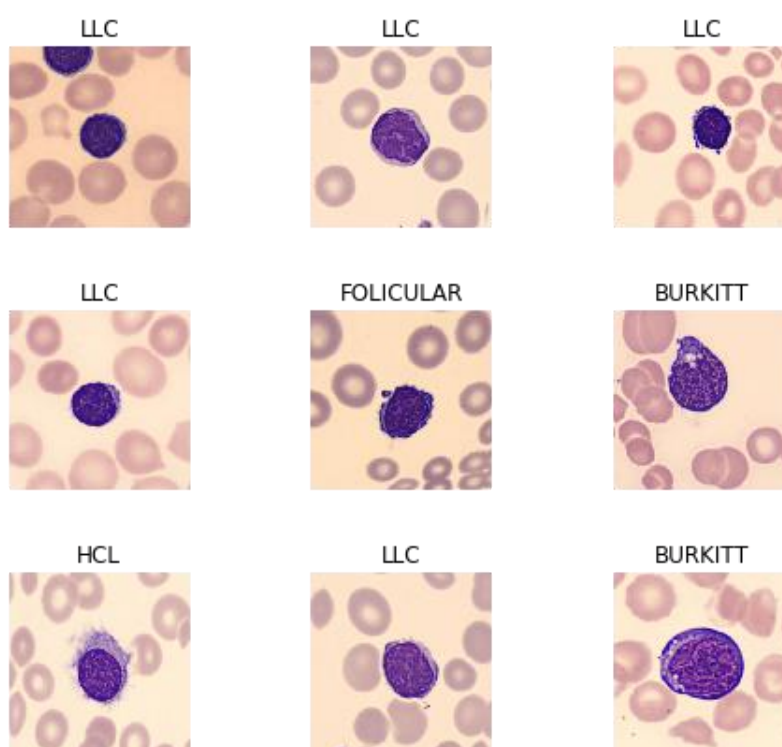


Figura 5.3. Relación de las imágenes con su etiqueta de clase.

A continuación se entrena el modelo utilizando una ResNet34 (ver Figura 5.4). Para ello, se utiliza la instrucción `create_cnn` (85). Este método crea un objeto `Learner` a partir de un objeto `data` y se le da una arquitectura `base_arch` (que en este caso es la ResNet34). Hay que añadir también las métricas `metrics` que se deseen utilizar (por ejemplo, `error_rate`).

```
data = ImageDataBunch.from_name_re(path, fnames, pat,
ds_tfms=get_transforms(), size=363).normalize(imagenet_stats)
learn = cnn_learner(data, models.resnet34, metrics=error_rate)
```

Figura 5.4. Sección de código 1: `data` y `cnn_learner`.

A continuación se utiliza `fit_one_cycle` (86) para entrenar el modelo (ver Figura 5.5) siguiendo la técnica de *1 policy cycle* (87), que permite entrenar un modelo de manera muy rápida empleando superconvergencia (88) (véase Tabla 6.1). La instrucción `ClassificationInterpretation` (89) proporciona una matriz de confusión de los resultados obtenidos y permite visualizar las imágenes peor clasificadas (`top_losses`) (véase Figura 6.1).

```
learn.fit_one_cycle(4)
interp = ClassificationInterpretation.from_learner(learn)
losses, idxs = interp.top_losses()
len(data.valid_ds) == len(losses) == len(idxs)

interp.plot_top_losses(9, figsize=(15,11), heatmap=True)

interp.plot_confusion_matrix(figsize=(9,9), dpi=60, normalize=False)
```

Figura 5.5. Sección de código 2: `fit_one_cycle`, `ClassificationInterpretation` y `top_losses`.

Se observa que el modelo funciona relativamente bien, por lo que se realiza un descongelamiento de todas las capas neuronales mediante `unfreeze` (90) y se prosigue entrenando el modelo. `Unfreeze` permite mantener los pesos de las primeras capas por lo que el modelo retiene mejor las representaciones básicas aprendidas. Tal y como se puede observar en la Tabla 6.3, el error cometido por el modelo ha disminuido. Este proceso se conoce como *fine tuning*, donde un modelo previamente entrenado se vuelve a entrenar para que realice una tarea similar modificando ciertos parámetros para mejorar su desempeño.

`Lr_find` (91) permite encontrar el *learning rate* más adecuado para el modelo. El *learning rate* es un hiperparámetro que controla cuanto se están ajustando los pesos de la red neuronal con respecto al gradiente de la función de pérdida (92).

```
learn.lr_find()
learn.recorder.plot()
```

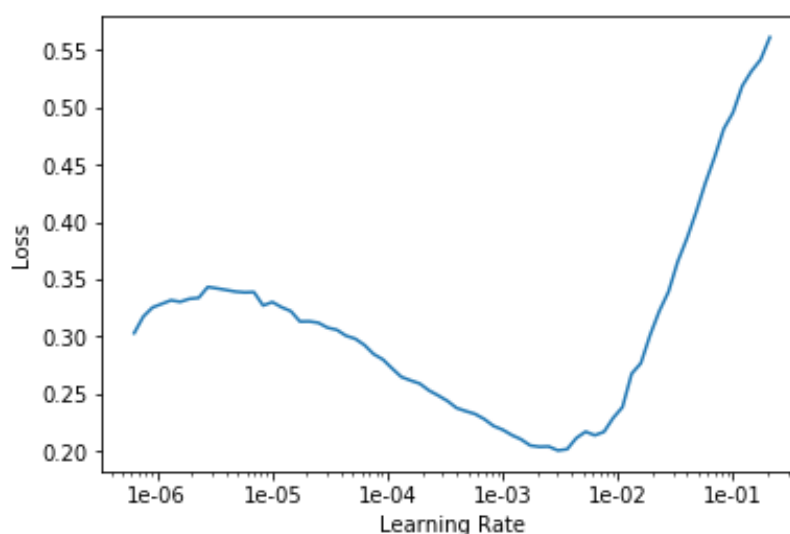


Figura 5.6. Función de pérdida vs LR del modelo del bloque 1.

Utilizando el *learning rate* (1×10^{-3}) encontrado (Figura 5.6) se vuelve a entrenar el modelo mediante `unfreeze` y utilizando `fit_one_cycle` durante 2 iteraciones (epoch) (ver Tabla 6.4).

Se utiliza nuevamente `ClassificationInterpretation` para encontrar la matriz de confusión (ver Tabla 6.2) y los `top_losses` (ver Figura 6.2) para poder comparar los nuevos resultados con los obtenidos anteriormente.

Se utiliza `show_results` (93) para visualizar las predicciones del modelo comparando la etiqueta inicial o *ground truth* y la etiqueta propuesta por el modelo.

5.3.2. Bloque 2 (ResNet50 convencional)

Se sigue la misma metodología que en el bloque 1 pero se utiliza una ResNet50 para entrenar el modelo para ver si se obtiene una precisión mayor.

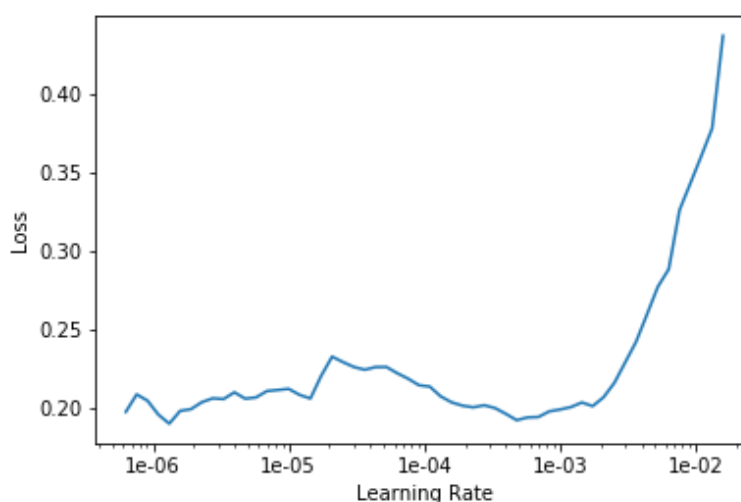


Figura 5.7. Función de pérdida vs LR del modelo del bloque 2.

Se utiliza el *learning rate* (1×10^{-3}) obtenido en la Figura 5.7 para encontrar los resultados de la Tabla 6.9.

5.3.3. Bloque 3 (SNN con arquitectura ResNet34 sin pre-entrenamiento)

Antes de empezar a trabajar, el dataset debe ser convertido a un objeto *DataBunch*, y en el caso de clasificación de imágenes debe convertirse a una subclase de *ImageDataBunch* (94). Tal y como se ha comentado en los bloques anteriores, la instrucción *pat* permite extraer la etiqueta de la clase de imagen de su propio nombre. A partir de esto y utilizando *fnames* se crea una lista del dataset mediante *ImageDataBunch.from_name_re* (95).

Se utiliza *to_df* para convertir el conjunto de datos a *dataframe*, que permite organizar los datos en una hoja de datos de la siguiente manera:

	Image	Id
0	BURKITT_32.jpg	BURKITT
1	LLC_6.jpg	LLC
2	FOLICULAR_20.jpg	FOLICULAR
3	BURKITT_16.jpg	BURKITT
4	HCL_21.jpg	HCL

Figura 5.8. Primeras cinco filas del dataframe.

Se observa en la Figura 5.8 que se han relacionado las imágenes en si (columna *Image*) con su etiqueta o clase (*Id*). A continuación se duplica el *dataframe* entero para generar un par positivo y un par

negativo de cada imagen (manteniendo un 50% de pares positivos y un 50% de pares negativos para la SNN) y se le añade una columna *target* con valores cero o uno que corresponden al *dataframe* original y al *dataframe* duplicado respectivamente. El *dataframe* pasa a tener unas dimensiones de 1002 filas y 3 columnas (el número de filas corresponde a las 501 imágenes duplicadas y las columnas corresponden a *Image*, *Id* y *target*).

	Image	Id	target
0	BURKITT_32.jpg	BURKITT	0
1	LLC_6.jpg	LLC	0
2	FOLICULAR_20.jpg	FOLICULAR	0
3	BURKITT_16.jpg	BURKITT	0
4	HCL_21.jpg	HCL	0

Figura 5.9. Primeras cinco filas del dataframe duplicado.

La librería de fast.ai permite crear el ítem de clase *ItemBase* y *ItemList* (ver Figura 5.10) (96) para SNN, que en este caso los ítems corresponden a *Image*. Se utilizan métodos incorporados en la librería como *from_df* que permiten instanciar la *ImageList* correspondiente. Se define *get()* *func* para conseguir pares de imágenes similares y pares de imágenes diferentes basados en un índice de valor, razón por la que se duplica el *dataframe* inicialmente. Esta función principalmente conecta la *ImageList* con la imagen *ItemBase* mediante el retorno del ítem conseguido como la clase de *ItemBase*. Estos pares de imágenes pueden observarse en la Figura 6.7.

```
mean, std = torch.tensor(imagenet_stats)

class SiamImage(ItemBase):
    def __init__(self, img1, img2):
        self.img1, self.img2 = img1, img2
        self.obj, self.data = (img1, img2), [(img1.data-mean[...,None,None])/std[...None,None], (img2.data-mean[...None,None])/std[...None,None]]

    def apply_tfms(self, tfms,*args, **kwargs):
        self.img1 = self.img1.apply_tfms(tfms, *args, **kwargs)
        self.img2 = self.img2.apply_tfms(tfms, *args, **kwargs)
        self.data = [(self.img1.data-mean[...None,None])/std[...None,None], (self.img2.data-mean[...None,None])/std[...None,None]]
        return self

    def __repr__(self): return f'{self.__class__.__name__} {self.img1.shape, self.img2.shape}'

    def to_one(self):
        return Image(mean[...None,None]+torch.cat(self.data,2)*std[...None,None])
```

```

class SiamImageItemList(ImageList):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)

    def __len__(self)->int: return len(self.items) or 1

    def get(self, i):
        match=1
        if i>=len(self.items)//2:
            match = 0
        fn = self.items[i]
        img1 = super().get(i)

        imgs = self.inner_df.Image.values
        ids = self.inner_df.Id.values
        wcls = ids[i]
        simgs = imgs[ids == wcls]
        dimgs = imgs[ids != wcls]
        if len(simgs)==1 and match==1:
            fn2=fn
        else:
            while True:
                np.random.shuffle(simgs)
                np.random.shuffle(dimgs)
                if simgs[0] != fn:
                    fn2 = [simgs[0] if match==1 else dimgs[0]][0]
                    break
            fn2 = self.items[np.where(imgs==fn2)[0][0]]
        img2 = super().open(fn2)
        return SiamImage(img1, img2)

    def reconstruct(self, t): return SiamImage(mean[... ,None,None]+t[0]*s
td[... ,None,None], mean[... ,None,None]+t[1]*std[... ,None,None])

    def show_xys(self, xs, ys, figsize:Tuple[int,int]=(9,10), **kwargs):
        rows = int(math.sqrt(len(xs)))
        fig, axs = plt.subplots(rows,rows,figsize=figsize)
        for i, ax in enumerate(axs.flatten() if rows > 1 else [axs]):
            xs[i].to_one().show(ax=ax, y=ys[i], **kwargs)
        plt.tight_layout()

```

Figura 5.10. Sección de código 3: ItemBase y ImageList.

A continuación se realiza una normalización de las imágenes antes de pasar por la SNN. La normalización se consigue mediante una transformación de las imágenes a 224 x 224 píxeles. Esta normalización puede observarse a continuación en la Figura 5.11.

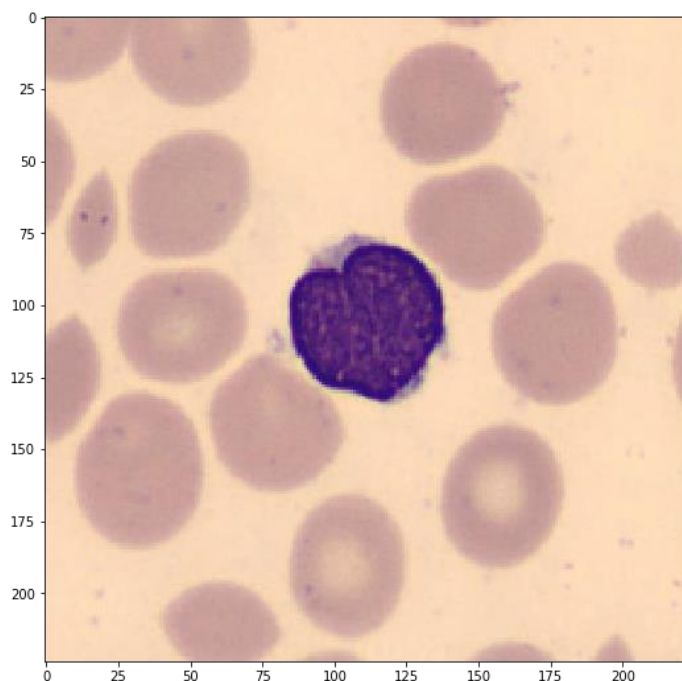


Figura 5.11. Imagen transformada.

Posteriormente se crea la SNN mediante la clase *SiameseNet* (ver Figura 5.12), utilizando la arquitectura ResNet34 creando dos subredes idénticas *body* y *head* mediante `Learner.create_body` y `Learner.create_head`. Se definen también las funciones `forward` y `get_embedding` que permiten obtener las salidas de la SNN.

```
from fastai.vision import learner

class SiameseNet(nn.Module):
    def __init__(self, arch=models.resnet34, lin_ftrs=[256, 128], emb_sz=
128, ps=0.5, bn_final=False):
        super(SiameseNet, self).__init__()
        self.arch, self.emb_sz = arch, emb_sz
        self.lin_ftrs, self.ps, self.bn_final = lin_ftrs, ps, bn_final
        self.body = learner.create_body(self.arch, True, learner.cnn_conf
ig(self.arch) ['cut'])
        self.head = learner.create_head(num_features_model(self.body) * 1
, self.emb_sz, self.lin_ftrs, self.ps, self.bn_final)
        self.cnn = nn.Sequential(self.body, self.head)

    def forward(self, x1, x2):
        output1 = self.cnn(x1)
        output2 = self.cnn(x2)
        return output1, output2

    def get_embedding(self, x):
        return self.cnn(x)
```

Figura 5.12. Sección de código 4: SiameseNet, forward y get_embedding.

Se utiliza *Contrastive Loss* como función de pérdida de la red neuronal (ver Figura 5.13).

```
class ContrastiveLoss(nn.Module):

    def __init__(self, margin=5.):
        super(ContrastiveLoss, self).__init__()
        self.margin = margin

    def forward(self, ops, target, size_average=True):
        op1, op2 = ops[0], ops[1]
        dist = F.pairwise_distance(op1, op2)
        pdist = dist*target
        ndist = dist*(1-target)
        loss = 0.5* ((pdist**2) + (F.relu(self.margin-ndist)**2))
        return loss.mean() if size_average else losses.sum()
```

Figura 5.13. Sección de código 5: ContrastiveLoss.

Esta función de pérdida basada en distancias permite que las imágenes similares sean incrustadas o *embedded* juntas y que las imágenes distintas se alejen. Se puede observar más gráficamente el objetivo de la función de pérdida *Contrastive Loss* en la Figura 5.14.

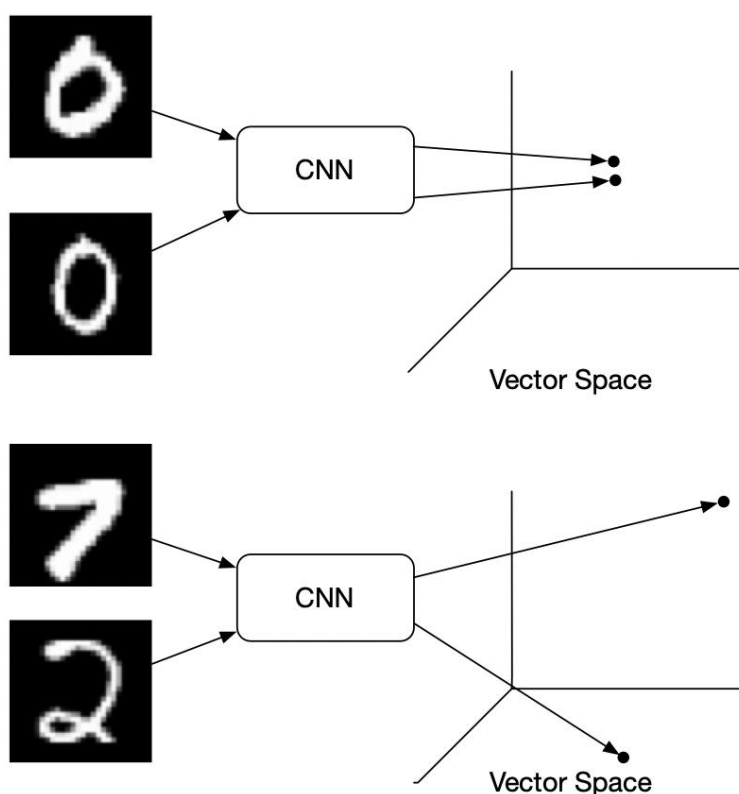


Figura 5.14. Objetivo de la función de pérdida Contrastive Loss (97).

Entonces se instancia el modelo y se entrena del mismo modo que en las secciones 5.3.1 y 5.3.2, es decir, se entrena utilizando `fit_one_cycle`, se realiza *fine tuning* de los parámetros, se encuentra el *learning rate* (1×10^{-5}) (ver Figura 5.15) y se re-entrena el modelo.

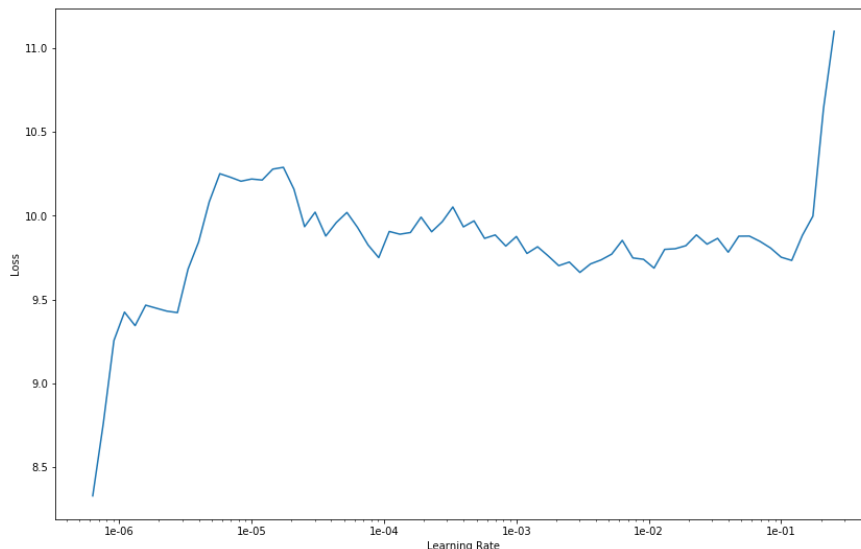


Figura 5.15. Función de pérdida vs LR del modelo del bloque 3.

Se utiliza la función `extract_embedding` (ver Figura 5.16) para separar los pares de imágenes y extraer las características o *features* de cada imagen. En la Figura 6.8 se observa gráficamente la posición espacial de cada una de las imágenes. Estas características se acoplan en un *dataframe* llamado *features* junto a la etiqueta de cada imagen. Para cada imagen existen 128 características.

```
def extract_embedding(dl, ds, mdl):
    mdl.eval()
    with torch.no_grad():
        preds = torch.zeros((len(ds), 128))
        start=0
        for cnt, (x, y) in enumerate(dl, start=0):
            size = x[0].shape[0]
            preds[start:start+size, :] = model.get_embedding(x[0])
            start+= size
        return preds
```

Figura 5.16. Sección de código 6: Extract_embedding.

A continuación se define la función `oneshot` (ver Figura 5.18) que permite calcular la distancia euclídea entre dos imágenes pasadas por la SNN. Esta distancia permite conocer si las dos imágenes pasadas por la función pertenecen a la misma clase o no (véase de la Tabla 6.15 a la Tabla 6.19). Para encontrar la distancia se ha realizado el promedio de las distancias encontradas para tres ejemplos de cada clase. La función `oneshot` tiene como entrada dos imágenes (ver Figura 5.17) y retorna como

salida un tensor distancia. En cada caso, se normaliza la distancia calculada en la función oneshot tomando como referencia la propia clase; todos los resultados de la función oneshot quedan recogidos en el capítulo 6.8.

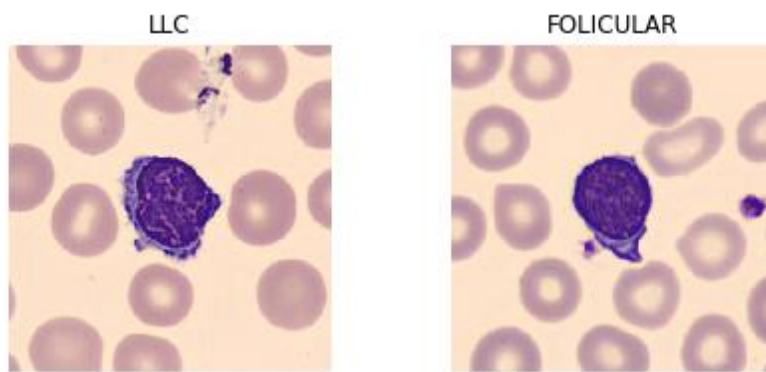


Figura 5.17. Ejemplo de entradas para la función oneshot.

```
def oneshot(model, img1, img2):  
    output1, output2 = model(img1.cuda(), img2.cuda())  
    euclidean_distance = F.pairwise_distance(output1, output2)  
    return euclidean_distance
```

Figura 5.18. Sección de código 7: oneshot.

5.3.4. Bloque 4 (SNN con arquitectura ResNet50 sin pre-entrenamiento)

Se sigue la misma metodología que en el bloque 3 (sección 5.3.3) pero se utiliza una ResNet50 como arquitectura base de la red neuronal para el modelo.

Se utiliza el *learning rate* (3×10^{-5}) obtenido en la Figura 5.19 para encontrar los resultados de la Tabla 6.22.

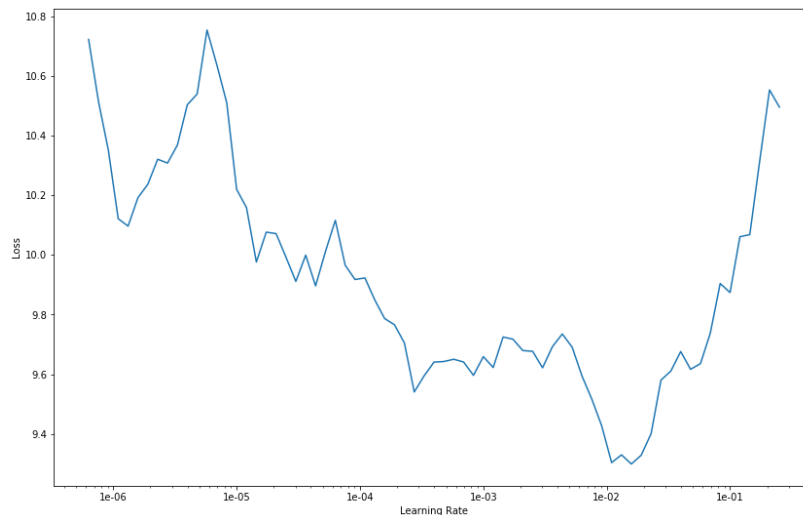


Figura 5.19. Función de pérdida vs LR del modelo del bloque 4.

En la Figura 6.9 se observan las características de las imágenes del bloque 4. Los resultados de la función oneshot quedan recogidos de la Tabla 6.23 a la Tabla 6.27.

5.3.5. Bloque 5 (SNN con arquitectura ResNet34 con pre-entrenamiento)

Se sigue la misma metodología que en el bloque 3 (sección 5.3.3) pero se utilizan los pesos de la red neuronal del bloque 1 (sección 5.3.1) para que la SNN de este bloque esté ya pre-entrenada. Para ello es necesario exportar el objeto *pesos_dic* del bloque 1 mediante el módulo *pickle* al bloque 5. Se utiliza la instrucción `Learn.model.state_dict()` para guardar estos pesos en un objeto *dictionary* (98) de la siguiente manera:

```
pretrained_dict = pesos_dic
modelo = models.resnet34(pretrained=True)
model_dict = modelo.state_dict()

pretrained_dict = {k: v for k, v in pretrained_dict.items() if k in model_dict}
model_dict.update(pretrained_dict)
modelo.load_state_dict(model_dict)

import pickle
pesos_dic
pickle_out = open("dict.pickle", "wb")
pickle.dump(pesos_dic, pickle_out)
pickle_out.close()
```

Figura 5.20. Sección de código 8: Guardado de pesos en un diccionario.

Puede observarse el entrenamiento del modelo en la Tabla 6.28. Utilizando el *learning rate* (1×10^{-5}) encontrado en la Figura 5.21 se obtienen los resultados de la Tabla 6.29 y de la Tabla 6.30.

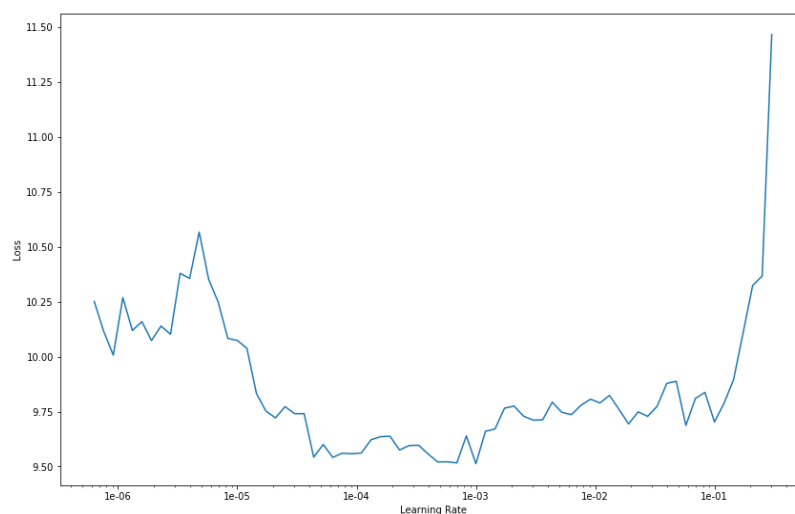


Figura 5.21. Función de pérdida vs LR del modelo del bloque 5.

En la Figura 6.10 se observan las características de las imágenes del bloque 5. Los resultados de la función oneshot quedan recogidos de la Tabla 6.31 a la Tabla 6.35.

5.3.6. Bloque 6 (SNN con arquitectura ResNet50 con pre-entrenamiento)

Se sigue la misma metodología que en el bloque 4 (sección 5.3.4) pero se utilizan los pesos de la red neuronal del bloque 2 (sección 5.3.2) para pre-entrenar la SNN de este bloque.

Se observa el entrenamiento inicial del modelo en la Tabla 6.36. Utilizando el *learning rate* (9×10^{-5}) encontrado en la Figura 5.22 se obtienen los resultados de la Tabla 6.37 y de la Tabla 6.38.

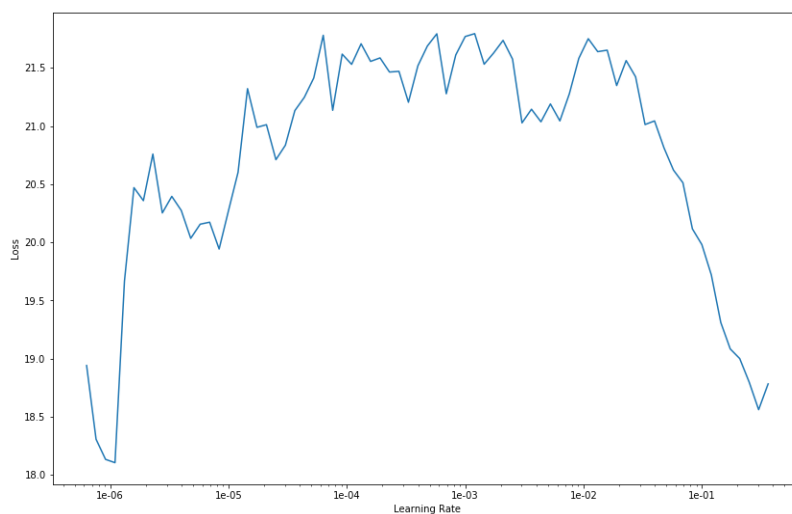


Figura 5.22. Función de pérdida vs LR del modelo del bloque 6.

En la Figura 6.11 se observan las características de las imágenes del bloque 6. Los resultados de la función oneshot quedan recogidos de la Tabla 6.39 a la Tabla 6.43.

6. Resultados

Se puede observar en las matrices de confusión (y por el número de imágenes de entrada) que en los bloques 1 y 2 se ha realizado una partición 80/20 de las imágenes iniciales (se han utilizado un 80% de las imágenes para el conjunto de *training* y un 20% para el conjunto de *validation*). Cabe destacar que en los modelos 3, 4, 5 y 6 también se ha utilizado una partición 80/20 para entrenar y validar los modelos.

6.1. Resultados del modelo ResNet34 convencional (Bloque 1)

En este capítulo se encuentran los resultados obtenidos siguiendo la metodología de la sección 5.3.1. En la Tabla 6.1 se observa el primer entrenamiento del modelo del bloque 1 durante 4 epoch. Las columnas *train_loss* y *valid_loss* permiten visualizar si el modelo está sufriendo *underfitting* u *overfitting*. En el caso que el valor de *valid_loss* fuera mucho más elevado que el valor de *train_loss* se puede afirmar que se está produciendo *overfitting*. En el caso que el valor de *valid_loss* sea algo mayor que el valor de *train_loss* se puede afirmar que se está produciendo un grado menor de *overfitting* (no necesariamente indica que sea malo). En el caso de que el valor de *valid_loss* sea algo menor que el valor de *train_loss* indica que se está produciendo un grado menor de *underfitting*. En el caso de que el valor de *valid_loss* sea mucho menor que el valor de *train_loss* se puede afirmar que se está produciendo *underfitting*. El objetivo principal de cualquier modelo es reducir al máximo el valor de *valid_loss*.

Train_loss hace referencia al valor de la función de pérdida del conjunto de entrenamiento y *valid_loss* hace referencia al valor de la función de pérdida del conjunto de validación.

Tabla 6.1. Entrenamiento del modelo del bloque 1 utilizando *fit_one_cycle* durante 4 epoch.

epoch	train_loss	valid_loss	error_rate
0	1,650649	1,257648	0,410000
1	1,153275	0,694469	0,260000
2	0,869613	0,483755	0,170000
3	0,705703	0,404148	0,190000

Se observa que en el último epoch el valor de *valid_loss* está algo por debajo del valor de *train_loss*, por lo que el modelo está sufriendo un grado menor de *underfitting* y se obtiene un *error rate* del 19%. En el caso del penúltimo epoch se obtiene un *error rate* del 17%.

En la Tabla 6.2 se visualiza la matriz de confusión normalizada para el primer entrenamiento del modelo del bloque 1:

Tabla 6.2. Matriz de confusión normalizada de la clasificación inicial del bloque 1.

Actual	Burkitt	88%	3%		3%	6%
	Folicular		97%			3%
	HCL		14%	57%	29%	
	LCM		17%	8%	67%	8%
	LLC		15%		39%	46%
		Burkitt	Folicular	HCL	LCM	LLC
Predicted						

A continuación, en la Figura 6.1, se observan las imágenes que han sido peor clasificadas por el modelo inicialmente. Encima de cada imagen se observa (de izquierda a derecha): la clase predicha por el modelo, la clase real de la imagen, la pérdida de esta y la probabilidad con la que se ha hecho dicha clasificación.

Dado que se realizan varias clasificaciones para un mismo modelo, solo se explicarán los *top_losses* de la clasificación final de cada modelo.

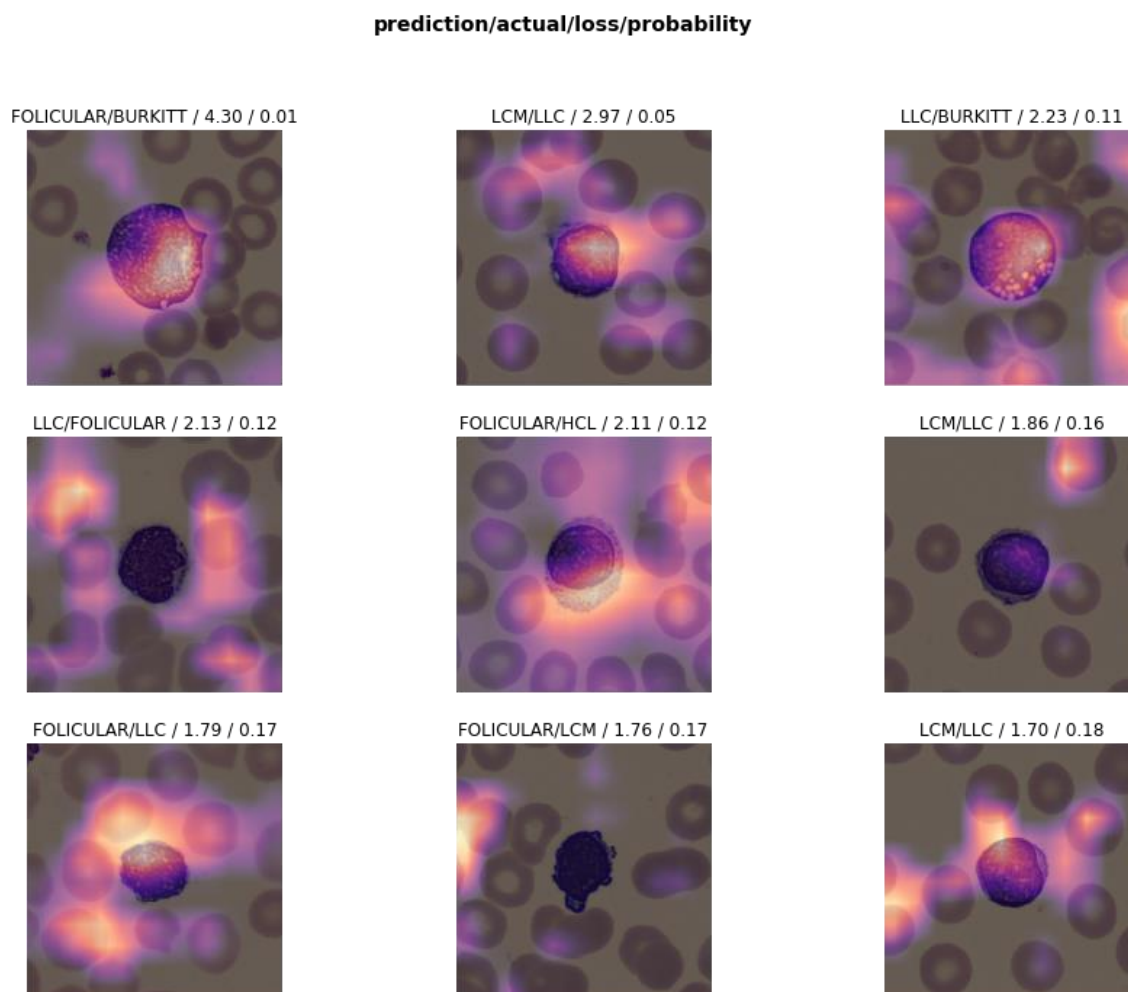


Figura 6.1. Top_losses de la clasificación inicial del bloque 1.

A continuación se realiza *fine tuning* de los parámetros del modelo y se entrena nuevamente el modelo para encontrar los resultados recogidos en la Tabla 6.3 y en la Tabla 6.4.

Tabla 6.3. Entrenamiento del modelo del bloque 1 utilizando fit_one_cycle durante 1 epoch después del unfreeze.

epoch	train_loss	valid_loss	error_rate
0	0,333999	0,279660	0,090000

Tabla 6.4. Entrenamiento del modelo del bloque 1 utilizando `fit_one_cycle` durante 2 epoch después de encontrar el learning rate.

epoch	train_loss	valid_loss	error_rate
0	0,265609	0,244069	0,080000
1	0,260747	0,280050	0,070000

Después de re-entrenar el modelo se observa en la Tabla 6.4 como se ha reducido el valor de pérdida de *train_loss* y *valid_loss*. Se observa también que tienen un valor muy similar por lo que se afirma que el modelo no sufre ni *overfitting* ni *underfitting*. En el segundo epoch se alcanza un *error rate* del 7%.

En la matriz de confusión (ver Tabla 6.5) se observan más gráficamente los resultados finales obtenidos en el bloque 1.

Tabla 6.5. Matriz de confusión normalizada de la segunda clasificación del bloque 1.

Actual	Burkitt	100%				
	Folicular		97%		3%	
	HCL		14%	58%	14%	14%
	LCM	8%			92%	
	LLC		8%		8%	84%
		Burkitt	Folicular	HCL	LCM	LLC
Predicted						

Comparando la segunda clasificación con la clasificación inicial (ver Tabla 6.5 y Tabla 6.2) se observa una mejora en el rendimiento (sensibilidad) de clasificación de las cuatro clases a excepción de la clasificación de HCL. Aunque solamente se hayan clasificado siete imágenes de HCL se visualizan los *top_losses* (ver Figura 6.2) para entender los errores de clasificación.

prediction/actual/loss/probability

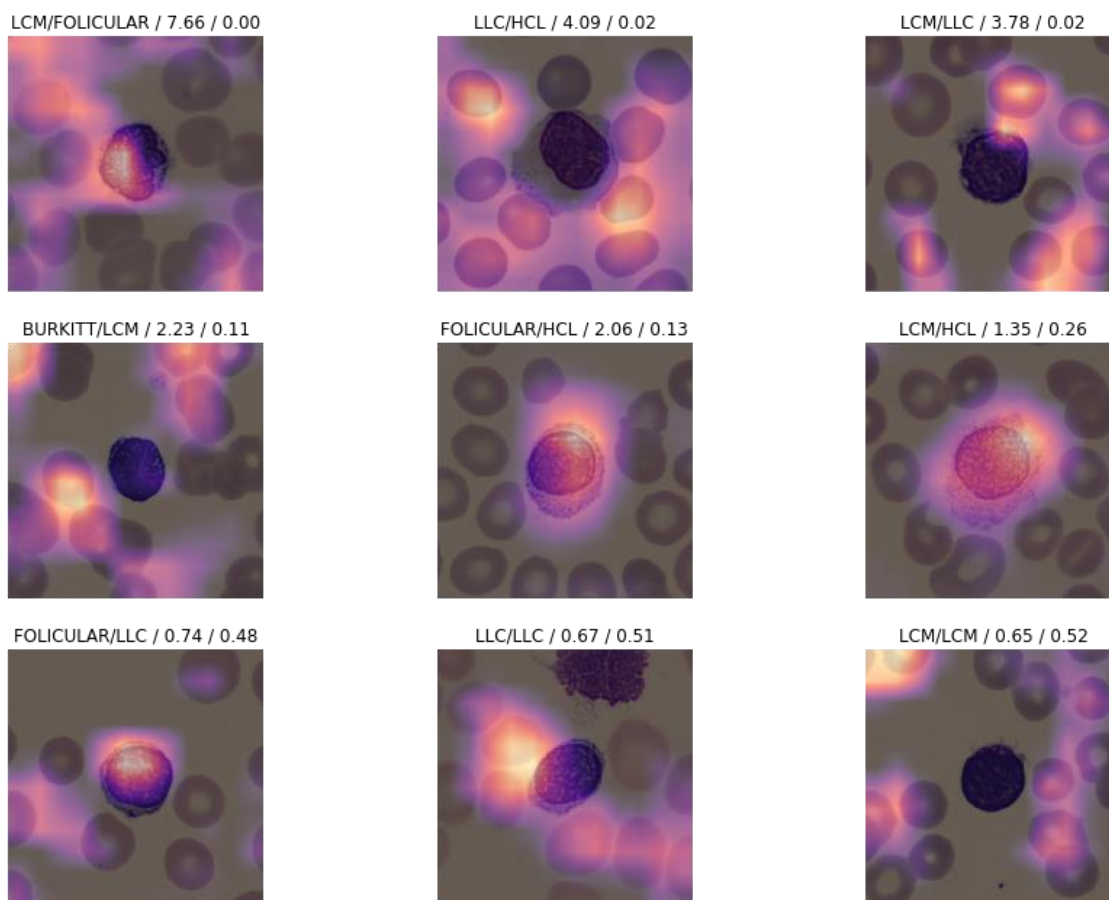


Figura 6.2. Top_losses de la segunda clasificación del bloque 1.

Se puede observar gracias al mapa de calor que las clasificaciones erróneas de HCL son debidas a que el propio modelo aprende a partir de las células sanguíneas circundantes y no del linfocito a estudiar. Se proponen opciones de mejora del modelo en el capítulo 8.

En la Figura 6.3 se observan los resultados finales del bloque 1. En las imágenes se compara la etiqueta real de la imagen (*Ground truth*) con la etiqueta propuesta por el modelo (*Predictions*).

6.2. Resultados del modelo ResNet50 convencional (Bloque 2)


UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
 Escola d'Enginyeria de Barcelona Est

En la Tabla 6.6 se observa el primer entrenamiento del modelo del bloque 2 durante 4 epoch. Se observa en el último epoch que el valor de *valid_loss* es algo inferior al valor de *train_loss*, por lo que el modelo sufre en un grado menor de *underfitting* y se obtiene un *error rate* del 8%.

Tabla 6.6. Entrenamiento del modelo del bloque 2 utilizando *fit_one_cycle* durante 4 epoch.

epoch	train_loss	valid_loss	error_rate
0	1,187658	1,319719	0,640000
1	0,741704	0,274770	0,090000
2	0,533010	0,223752	0,080000
3	0,421917	0,198464	0,080000

En la Tabla 6.7 se observa la matriz de confusión normalizada para el primer entrenamiento del bloque 2.

Tabla 6.7. Matriz de confusión normalizada de la clasificación inicial del bloque 2.

Actual	Burkitt	100%				
	Folicular		96%		4%	
	HCL			91%	9%	
	LCM	7%	7%		79%	7%
	LLC		28%			72%
		Burkitt	Folicular	HCL	LCM	LLC
Predicted						

A continuación, en la Figura 6.4, se observan las imágenes peor clasificadas por el modelo en su primer entrenamiento.

prediction/actual/loss/probability

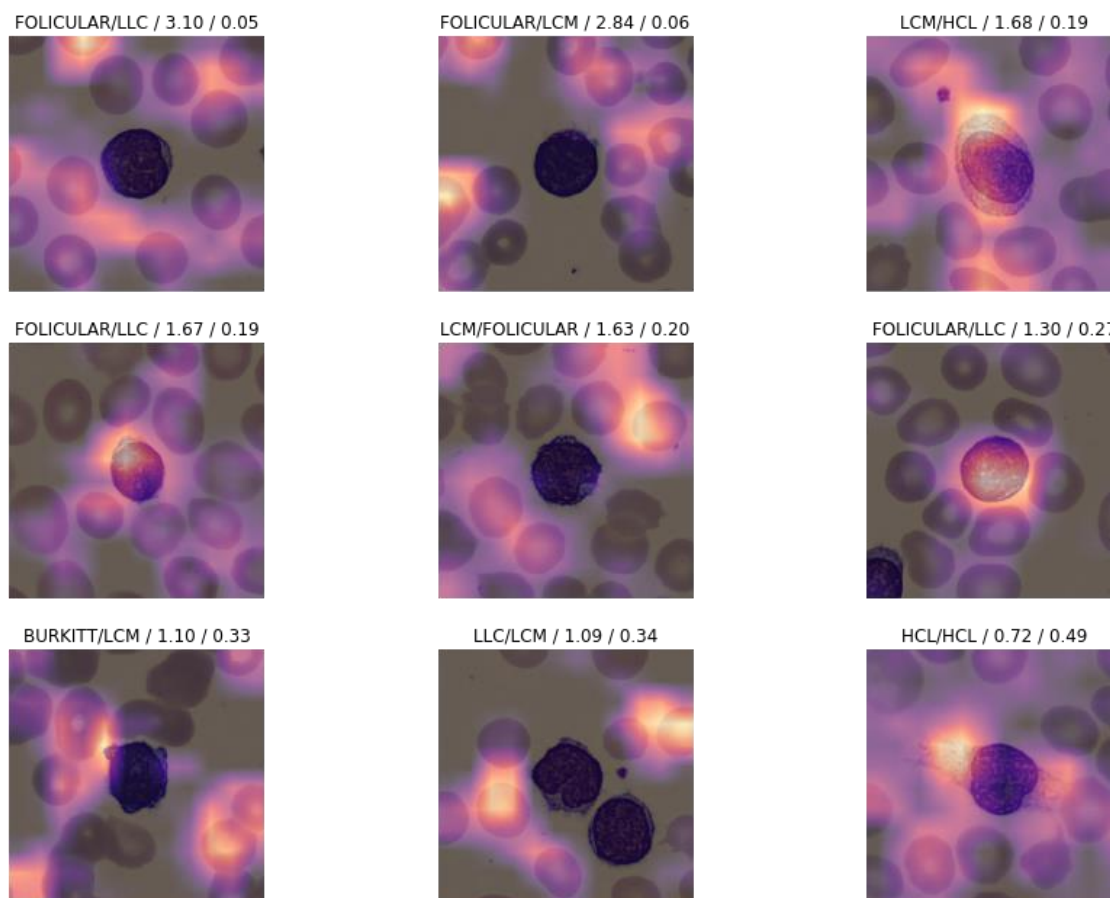


Figura 6.4. Top_losses de la clasificación inicial del bloque 2.

En la Tabla 6.8 y en la Tabla 6.9 se observan los siguientes entrenamientos del modelo después de haber realizado *fine tuning*.

Tabla 6.8. Entrenamiento del modelo del bloque 2 utilizando fit_one_cycle durante 1 epoch después del unfreeze.

epoch	train_loss	valid_loss	error_rate
0	0,395359	1,479071	0,270000

Tabla 6.9. Entrenamiento del modelo del bloque 2 utilizando fit_one_cycle durante 2 epoch después de encontrar el learning rate.

epoch	train_loss	valid_loss	error_rate
0	0,165928	0,096083	0,040000
1	0,129528	0,064555	0,020000

Después de re-entrenar el modelo se observa en la Tabla 6.9 que se han reducido los valores de *train_loss* y *valid_loss*. El valor de *valid_loss* es algo inferior al valor de *train_loss* pero no es considerado *underfitting* del modelo. Se alcanza un *error rate* del 2% en el segundo epoch.

En la matriz de confusión normalizada (Tabla 6.10) se observan más gráficamente los resultados finales obtenidos en el bloque 2.

Tabla 6.10. Matriz de confusión normalizada de la segunda clasificación del bloque 2.

Actual	Burkitt	100%				
	Folicular		100%			
	HCL		9%	91%		
	LCM				93%	7%
	LLC					100%
		Burkitt	Folicular	HCL	LCM	LLC
Predicted						

En esta segunda clasificación se observa una mejora en la clasificación de las clases Folicular (96% a 100%), LCM (79% a 93%) y LLC (72% a 100%). Solamente se han clasificado erróneamente dos imágenes en este bloque. El modelo ha confundido un HCL por un Folicular y un LCM por un LLC.

prediction/actual/loss/probability

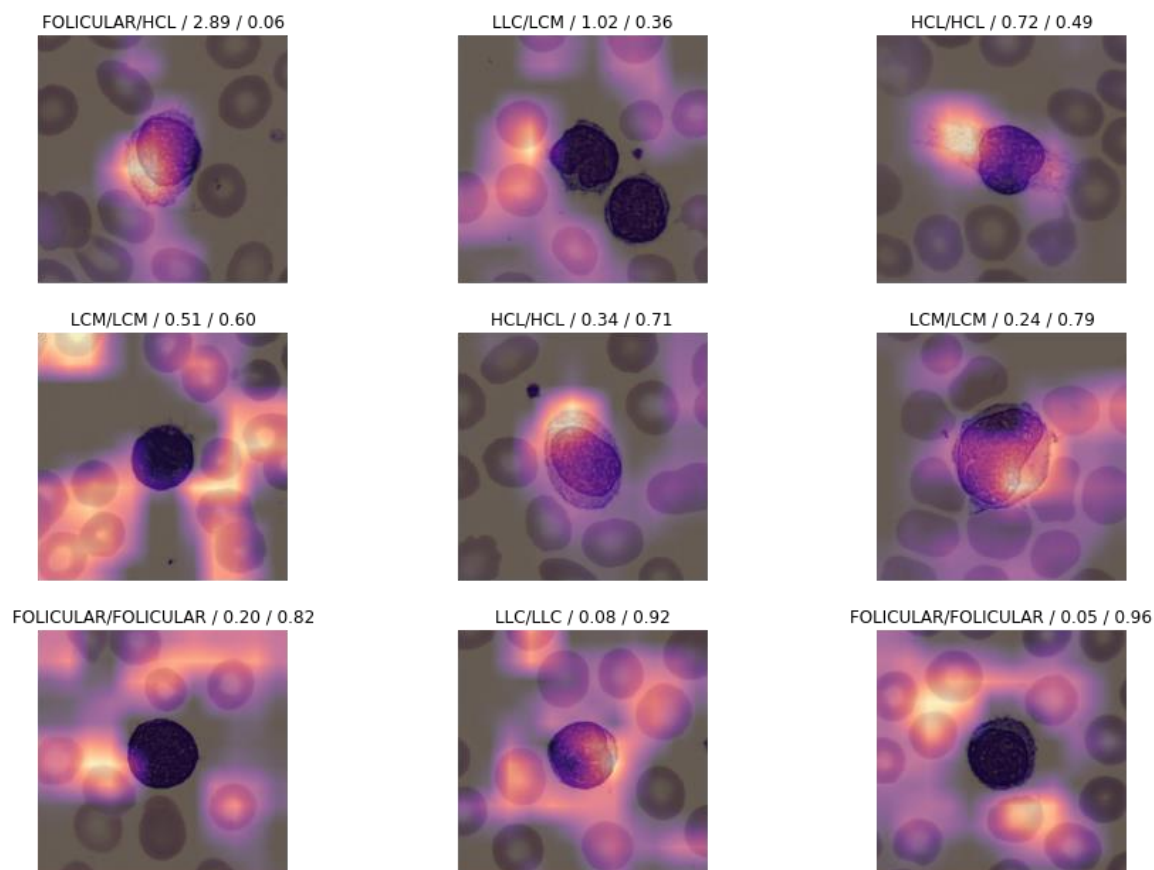


Figura 6.5. Top_losses de la segunda clasificación del bloque 2.

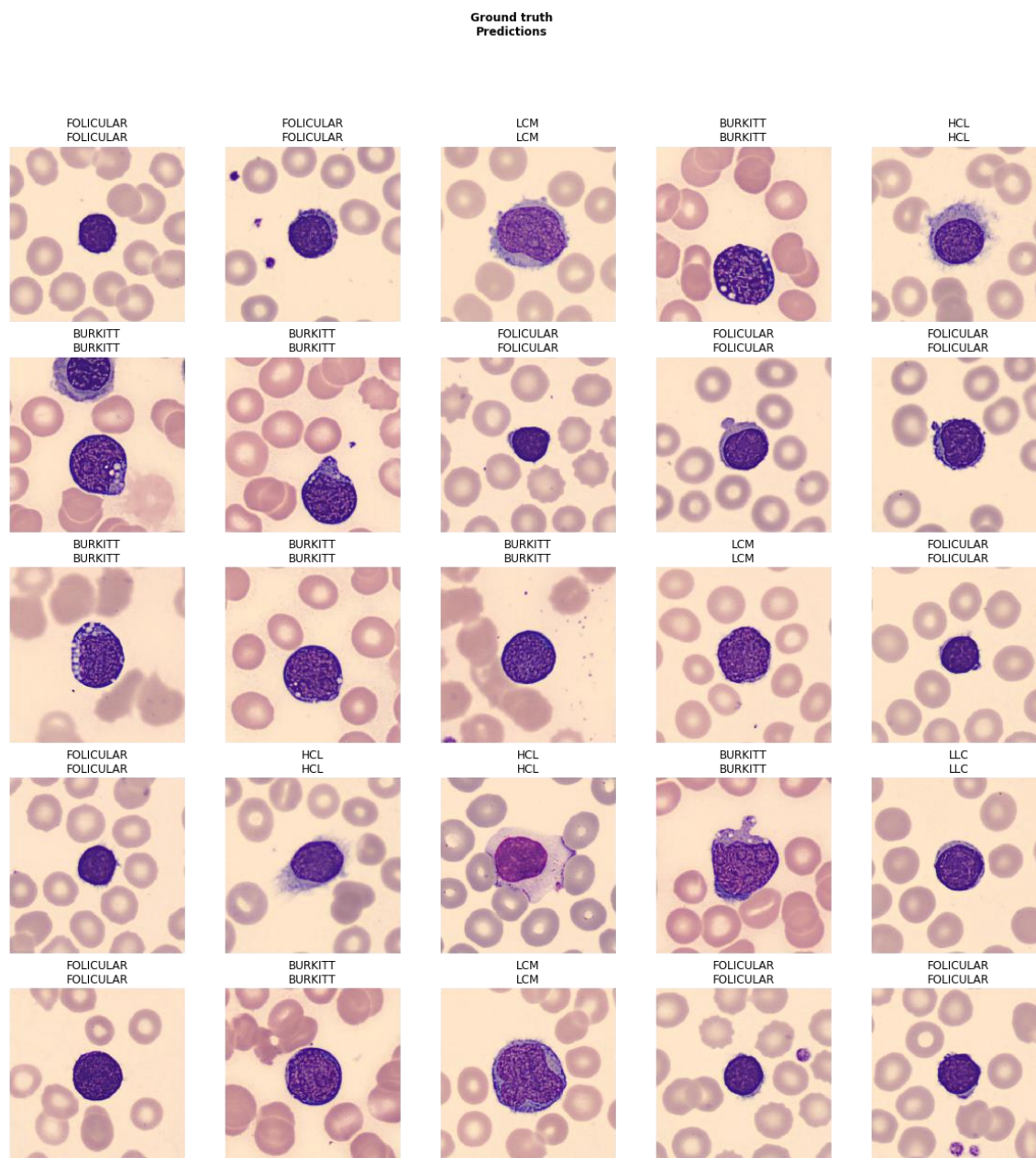


Figura 6.6. Show_results del bloque 2.

En la Figura 6.6 se observan los resultados finales del bloque 2.

6.3. Comparativa ResNet34 y ResNet50 convencionales

Utilizando las matrices de confusión y los datos obtenidos durante los entrenamientos de los bloques 1 y 2 se estudian las diferencias entre usar una ResNet34 y una ResNet50 (véase Tabla 6.11).

Tabla 6.11. Comparativa de los resultados de los bloques 1 y 2.

		Bloque 1 (ResNet34)	Bloque 2 (ResNet50)
Error rate	Inicial	19%	8%
	Final	7%	2%
Sensibilidad	Burkitt	100%	100%
	Folicular	97%	100%
	HCL	58%	91%
	LCM	92%	93%
	LLC	84%	100%

Hace falta destacar que aunque el modelo del bloque 1 haya obtenido una sensibilidad del 57% en la clasificación de HCL; no debe entenderse como un resultado estrictamente malo, puesto que el modelo solo ha utilizado 7 imágenes para la clasificación de HCL.

Se observa una mejora de sensibilidad en cada una de las cinco clases a clasificar utilizando una ResNet50, aunque también se incrementa el coste computacional al utilizar esta red neuronal.

6.4. Resultados del modelo SNN (ResNet34) sin pre-entrenamiento (Bloque 3)

En este capítulo se encuentran los resultados obtenidos siguiendo la metodología del capítulo 5.3.3. En la Figura 6.7 se puede observar la generación de pares de imágenes mediante la *ItemList* e *ItemBase* creados.

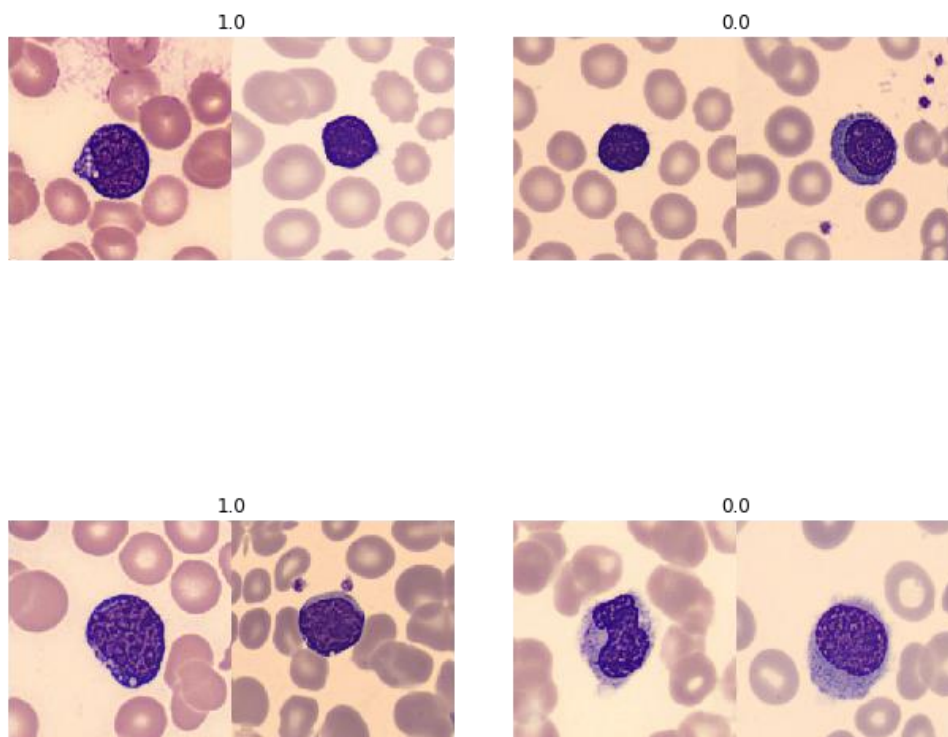


Figura 6.7. Generación de pares de imágenes.

En la Tabla 6.12 se observa el primer entrenamiento del modelo del bloque 3 durante 5 epoch. A diferencia de los dos capítulos anteriores, no aparece la columna *error_rate*, esto es debido a que las SNN no realizan una clasificación normal como se ha comentado, por lo que no tiene sentido intentar encontrar este tipo de error.

Tabla 6.12. Entrenamiento del modelo del bloque 3 utilizando fit_one_cycle durante 5 epoch.

epoch	train_loss	valid_loss
0	9,711779	12,998087
1	9,422467	11,821211
2	9,447232	10,944741
3	9,448924	11,128979
4	9,438851	11,205179

Se observa que el valor de *valid_loss* está por encima del valor de *train_loss*, por lo que el modelo sufre algo de *overfitting*. Se encuentra el valor más bajo de *valid_loss* en el tercer epoch.

A continuación se realiza *fine tuning* de los parámetros del modelo y se entrena nuevamente. Se observan los resultados en la Tabla 6.13 y en la Tabla 6.14.

Tabla 6.13. Entrenamiento del modelo del bloque 3 utilizando *fit_one_cycle* durante 1 epoch después del unfreeze.

epoch	train_loss	valid_loss
0	9,488190	11,013353

Tabla 6.14. Entrenamiento del modelo del bloque 3 utilizando *fit_one_cycle* durante 5 epoch después de encontrar el learning rate.

epoch	train_loss	valid_loss
0	9,422415	11,450194
1	9,318882	11,394958
2	9,399467	11,583862
3	9,355668	11,457361
4	9,434452	11,283150

Después de re-entrenar el modelo puede observarse que no se han conseguido reducir los valores de *train_loss* y *valid_loss*.

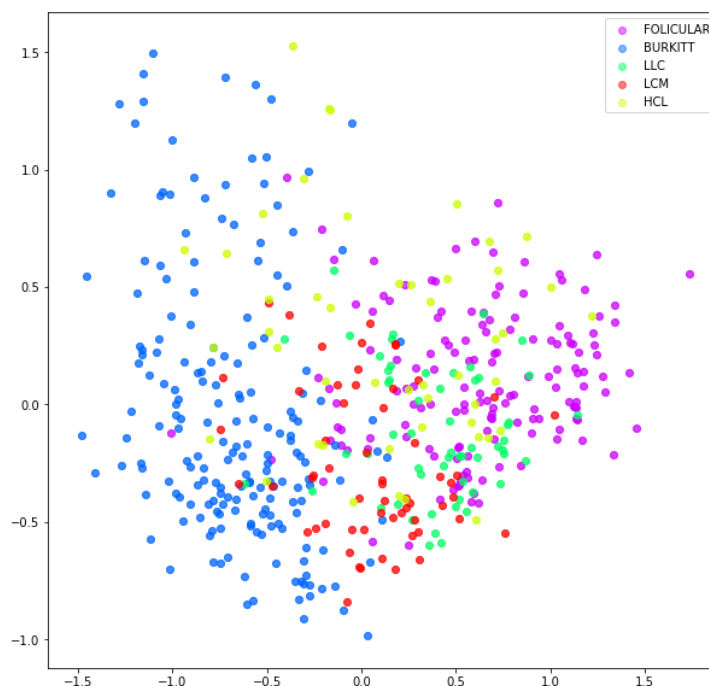


Figura 6.8. Disposición espacial de las dos primeras componentes principales de las imágenes del bloque 3.

En la Figura 6.8 pueden verse representadas espacialmente las dos primeras componentes principales de las imágenes del bloque 3.

En la Tabla 6.15 se observan los resultados de aplicar la función oneshot para la clase Burkitt. Para la misma clase se obtiene una distancia media de 0,4987 y para las demás clases una distancia mínima de 1,4473.

Tabla 6.15. Resultados función oneshot para la clase Burkitt (Bloque 3).

Clase	Clase a comparar	Distancia	Distancia normalizada
BURKITT	BURKITT	0,4987	100%
	FOLICULAR	1,8757	376%
	HCL	1,9728	396%
	LCM	1,4985	300%
	LLC	1,4473	290%

En la Tabla 6.16 se observan los resultados de aplicar la función oneshot para la clase Folicular. Para la misma clase se obtiene una distancia media de 1,0534 y para las demás clases una distancia mínima de 1,5582.

Tabla 6.16. Resultados función oneshot para la clase Folicular (Bloque 3).

Clase	Clase a comparar	Distancia	Distancia normalizada
FOLICULAR	BURKITT	2,2991	218%
	FOLICULAR	1,0534	100%
	HCL	1,5582	148%
	LCM	1,6920	161%
	LLC	1,6811	160%

En la Tabla 6.17 se observan los resultados de aplicar la función oneshot para la clase HCL. Para la misma clase se obtiene una distancia media de 1,0427 y para las demás clases una distancia mínima de 1,3356.

Tabla 6.17. Resultados función oneshot para la clase HCL (Bloque 3).

Clase	Clase a comparar	Distancia media	Distancia normalizada
HCL	BURKITT	1,7382	167%
	FOLICULAR	1,5582	149%
	HCL	1,0427	100%
	LCM	1,8073	173%
	LLC	1,3356	128%

En la Tabla 6.18 se observan los resultados de aplicar la función oneshot para la clase LCM. Para la misma clase se obtiene una distancia media de 0,9128 y para las demás clases una distancia mínima de 1,3075.

Tabla 6.18. Resultados función oneshot para la clase LCM (Bloque 3).

Clase	Clase a comparar	Distancia	Distancia normalizada
LCM	BURKITT	1,7105	187%
	FOLICULAR	1,3075	143%
	HCL	1,3120	144%
	LCM	0,9128	100%
	LLC	1,3547	148%

En la Tabla 6.19 se observan los resultados de aplicar la función oneshot para la clase LLC. Para la misma clase se obtiene una distancia media de 0,8908 y para las demás clases una distancia mínima de 1,2198.

Tabla 6.19. Resultados función oneshot para la clase LLC (Bloque 3).

Clase	Clase a comparar	Distancia	
LLC	BURKITT	1,6908	190%
	FOLICULAR	1,2198	137%
	HCL	1,2206	137%
	LCM	1,3486	151%
	LLC	0,8908	100%

6.5. Resultados del modelo SNN (ResNet50) sin pre-entrenamiento (Bloque 4)

En Tabla 6.20 se observa el primer entrenamiento del modelo del bloque 4 durante 5 epoch. Puede verse que el valor de *train_loss* está algo por debajo del valor de *valid_loss* por lo que el modelo sufre un grado menor de *overfitting*. Se encuentra el valor más bajo de *valid_loss* en el primer epoch con un valor de 10,591370.

Tabla 6.20. Entrenamiento del modelo del bloque 4 utilizando *fit_one_cycle* durante 5 epoch.

epoch	train_loss	valid_loss
0	10,714017	10,591370
1	9,737941	12,037616
2	9,505132	11,133039
3	9,577903	11,615798
4	9,431245	10,994203

A continuación se realiza *fine tuning* de los parámetros del modelo y se entrena nuevamente para encontrar los resultados recogidos en la Tabla 6.21 y en la Tabla 6.22.

Tabla 6.21. Entrenamiento del modelo del bloque 4 utilizando *fit_one_cycle* durante 1 epoch después del unfreeze.

epoch	train_loss	valid_loss
0	9,552506	10,530913

Tabla 6.22. Entrenamiento del modelo del bloque 4 utilizando fit_one_cycle durante 5 epoch después de encontrar el learning rate.

epoch	train_loss	valid_loss
0	9,386072	11,421229
1	9,552289	11,376604
2	9,575787	11,353097
3	9,550577	11,396743
4	9,494743	11,387245

Después de re-entrenar el modelo puede observarse como no se ha conseguido reducir los valores de *train_loss* y *valid_loss*.

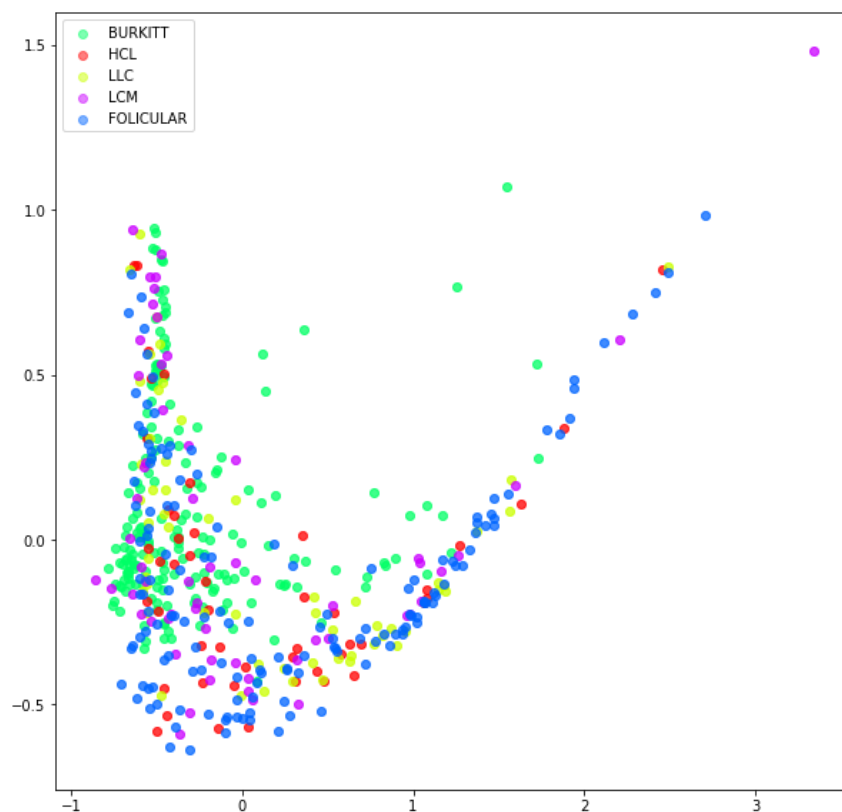


Figura 6.9. Disposición espacial de las dos primeras componentes principales de las imágenes del bloque 4.

En la Figura 6.9 pueden verse representadas espacialmente las dos primeras componentes principales de las imágenes del bloque 4.

En la Tabla 6.23 se observan los resultados de aplicar la función oneshot para la clase Burkitt. Para la misma clase se obtiene una distancia media de 0,0718 y para las demás clases una distancia mínima de 0,2730.

Tabla 6.23. Resultados función oneshot para la clase Burkitt (Bloque 4).

Clase	Clase a comparar	Distancia	Distancia normalizada
BURKITT	BURKITT	0,0718	100%
	FOLICULAR	0,3117	434%
	HCL	0,2931	408%
	LCM	0,2730	380%
	LLC	0,3198	445%

En la Tabla 6.24 se observan los resultados de aplicar la función oneshot para la clase Folicular. Para la misma clase se obtiene una distancia media de 0,0471 y para las demás clases una distancia mínima de 0,0851.

Tabla 6.24. Resultados función oneshot para la clase Folicular (Bloque 4).

Clase	Clase a comparar	Distancia	Distancia normalizada
FOLICULAR	BURKITT	0,2235	475%
	FOLICULAR	0,0471	100%
	HCL	0,1182	251%
	LCM	0,1431	304%
	LLC	0,0851	181%

En la Tabla 6.25 se observan los resultados de aplicar la función oneshot para la clase HCL. Para la misma clase se obtiene una distancia media de 0,0920 y para las demás clases una distancia mínima de 0,1309.

Tabla 6.25. Resultados función oneshot para la clase HCL (Bloque 4).

Clase	Clase a comparar	Distancia	Distancia normalizada
HCL	BURKITT	0,1452	158%
	FOLICULAR	0,1309	142%
	HCL	0,0920	100%
	LCM	0,1512	164%
	LLC	0,1359	148%

En la Tabla 6.26 se observan los resultados de aplicar la función oneshot para la clase LCM. Para la misma clase se obtiene una distancia media de 0,0432 y para las demás clases una distancia mínima de 0,0612.

Tabla 6.26. Resultados función oneshot para la clase LCM (Bloque 4).

Clase	Clase a comparar	Distancia	Distancia normalizada
LCM	BURKITT	0,2228	516%
	FOLICULAR	0,0612	142%
	HCL	0,1220	282%
	LCM	0,0432	100%
	LLC	0,0992	230%

En la Tabla 6.27 se observan los resultados de aplicar la función oneshot para la clase LLC. Para la misma clase se obtiene una distancia media de 0,0210 y para las demás clases una distancia mínima de 0,0691.

Tabla 6.27. Resultados función oneshot para la clase LLC (Bloque 4).

Clase	Clase a comparar	Distancia	Distancia normalizada
LLC	BURKITT	0,1245	593%
	FOLICULAR	0,1139	542%
	HCL	0,1070	510%
	LCM	0,0691	329%
	LLC	0,0210	100%

6.6. Resultados del modelo SNN (ResNet34) con pre-entrenamiento (Bloque 5)

En este capítulo se encuentran recogidos los resultados obtenidos siguiendo la metodología del capítulo 5.3.5. En la Tabla 6.28 se observa el primer entrenamiento del modelo del bloque 5 durante 2 epoch. El valor de *train_loss* está algo por debajo del valor de *valid_loss* por lo que el modelo sufre algo de *overfitting*.

Tabla 6.28. Entrenamiento del modelo del bloque 5 utilizando fit_one_cycle durante 2 epoch.

epoch	train_loss	valid_loss
0	9,320746	10,367023
1	9,465702	10,761040

A continuación se realiza *fine tuning* de los parámetros del modelo y se entrena nuevamente. Se recogen los resultados en la Tabla 6.29 y en la Tabla 6.30.

Tabla 6.29. Entrenamiento del modelo del bloque 5 utilizando fit_one_cycle durante 4 epoch después del unfreeze.

epoch	train_loss	valid_loss
0	9,465879	10,880427
1	9,547116	11,007293
2	9,697095	11,020259
3	9,452457	10,977212

Tabla 6.30. Entrenamiento del modelo del bloque 5 utilizando fit_one_cycle durante 4 epoch después de encontrar el learning rate.

epoch	train_loss	valid_loss
0	9,349730	10,920542
1	9,457721	10,928866
2	9,537915	10,996984
3	9,571998	10,970201

Después de re-entrenar el modelo puede observarse que no se han conseguido reducir los valores de *train_loss* y *valid_loss*. Se obtiene el valor más bajo de *valid_loss* en el primer epoch con un valor de 10,920542. El modelo sigue sufriendo algo de *overfitting*.

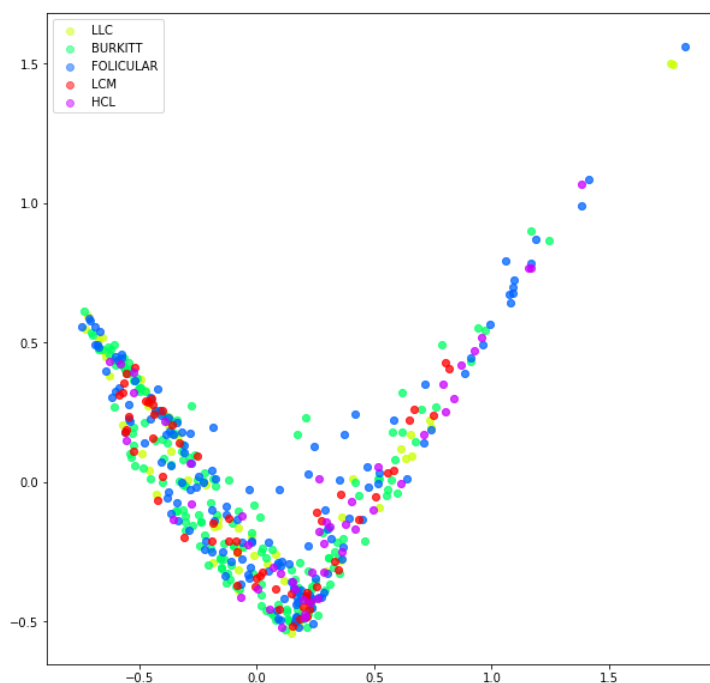


Figura 6.10. Disposición espacial de las dos primeras componentes principales de las imágenes del bloque 5.

En la Figura 6.10 pueden verse representadas espacialmente las dos primeras componentes principales de las imágenes del bloque 5.

En la Tabla 6.31 se observan los resultados de aplicar la función oneshot para la clase Burkitt. Para la misma clase se obtiene una distancia media de 0,0003 y para las demás clases una distancia mínima de 0,0012.

Tabla 6.31. Resultados función oneshot para la clase Burkitt (Bloque 5).

Clase	Clase a comparar	Distancia	Distancia normalizada
BURKITT	BURKITT	0,0003	100%
	FOLICULAR	0,0015	500%
	HCL	0,0012	400%
	LCM	0,0013	433%
	LLC	0,0021	700%

En la Tabla 6.32 se observan los resultados de aplicar la función oneshot para la clase Folicular. Para la misma clase se obtiene una distancia media de $6,1482 \times 10^{-5}$ y para las demás clases una distancia mínima de 0,0006.

Tabla 6.32. Resultados función oneshot para la clase Folicular (Bloque 5).

Clase	Clase a comparar	Distancia	Distancia normalizada
FOLICULAR	BURKITT	0,0008	1.301%
	FOLICULAR	$6,1482 \times 10^{-5}$	100%
	HCL	0,0007	1.139%
	LCM	0,0006	976%
	LLC	0,0007	1.139%

En la Tabla 6.33 se observan los resultados de aplicar la función oneshot para la clase HCL. Para la misma clase se obtiene una distancia media de $9,5990 \times 10^{-5}$ y para las demás clases una distancia mínima de 0,0004.

Tabla 6.33. Resultados función oneshot para la clase HCL (Bloque 5).

Clase	Clase a comparar	Distancia	Distancia normalizada
HCL	BURKITT	0,0014	1.458%
	FOLICULAR	0,0004	417%
	HCL	$9,5990 \times 10^{-5}$	100%
	LCM	0,0004	417%
	LLC	0,0007	729%

En la Tabla 6.34 se observan los resultados de aplicar la función oneshot para la clase LCM. Para la misma clase se obtiene una distancia media de $8,2485 \times 10^{-5}$ y para las demás clases una distancia mínima de 0,0004.

Tabla 6.34. Resultados función oneshot para la clase LCM (Bloque 5).

Clase	Clase a comparar	Distancia	Distancia normalizada
LCM	BURKITT	0,0017	2.061%
	FOLICULAR	0,0006	727%
	HCL	0,0004	485%
	LCM	$8,2485 \times 10^{-5}$	100%
	LLC	0,0004	485%

En la Tabla 6.35 se observan los resultados de aplicar la función oneshot para la clase LLC. Para la misma clase se obtiene una distancia media de $6,3971 \times 10^{-5}$ y para las demás clases una distancia mínima de 0,0007.

Tabla 6.35. Resultados función oneshot para la clase LLC (Bloque 5).

Clase	Clase a comparar	Distancia	Distancia normalizada
LLC	BURKITT	0,0009	1.407%
	FOLICULAR	0,0010	1.563%
	HCL	0,0009	1.407%
	LCM	0,0007	1.094%
	LLC	$6,3971 \times 10^{-5}$	100%

6.7. Resultados del modelo SNN (ResNet50) con pre-entrenamiento (Bloque 6)

En este capítulo se encuentran los resultados obtenidos siguiendo la metodología del capítulo 5.3.6. En la Tabla 6.36 se observa el primer entrenamiento del modelo del bloque 6 durante 2 epoch. El valor de *valid_loss* está por debajo del valor de *train_loss*, por lo que el modelo sufre *underfitting*.

Tabla 6.36. Entrenamiento del modelo del bloque 6 utilizando *fit_one_cycle* durante 2 epoch.

epoch	train_loss	valid_loss
0	50,477539	11,317726
1	26,443201	10,414110

A continuación se realiza *fine tuning* de los parámetros del modelo y se entrena nuevamente. Se observan los resultados en la Tabla 6.37 y en la Tabla 6.38.

Tabla 6.37. Entrenamiento del modelo del bloque 6 utilizando *fit_one_cycle* durante 4 epoch después del unfreeze.

epoch	train_loss	valid_loss
0	22,645445	10,571001
1	22,368364	10,259024
2	22,605049	10,187833
3	21,891701	10,266346

Tabla 6.38. Entrenamiento del modelo del bloque 6 utilizando `fit_one_cycle` durante 4 epoch después de encontrar el learning rate.

epoch	train_loss	valid_loss
0	21,605576	10,406405
1	21,502920	10,342107
2	20,051065	10,357159
3	20,775637	10,378439

Después de re-entrenar el modelo puede observarse que no se ha conseguido reducir el valor de *valid_loss*. Se ha conseguido reducir el valor de *train_loss*. El valor más bajo de *valid_loss* lo encontramos en el segundo epoch con un valor de 10,342107. El modelo sigue sufriendo *underfitting*.

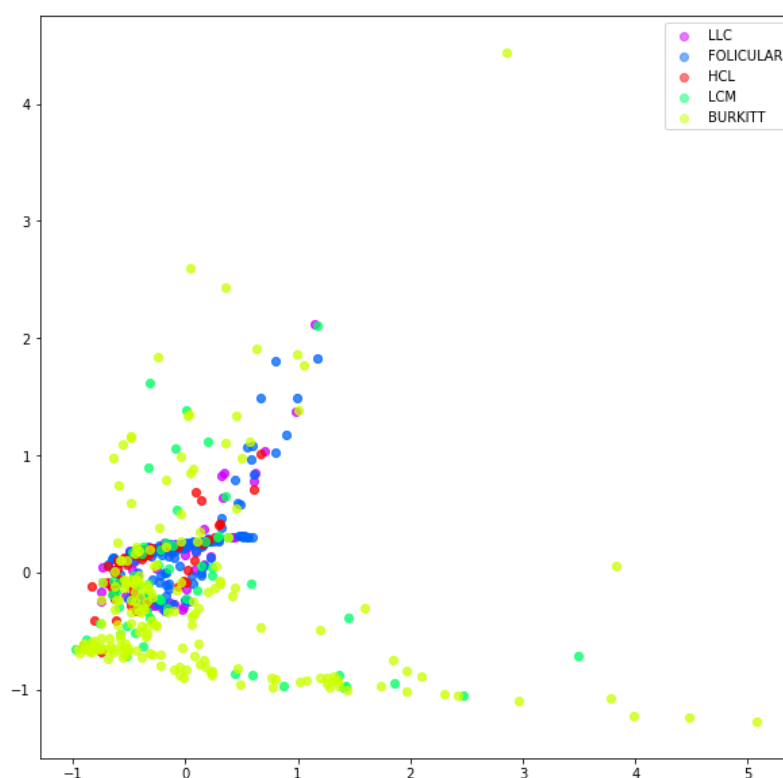


Figura 6.11. Disposición espacial de las dos primeras componentes principales de las imágenes del bloque 6.

En la Figura 6.11 pueden verse representadas espacialmente las dos primeras componentes principales de las imágenes del bloque 6.

En la Tabla 6.39 se observan los resultados de aplicar la función oneshot para la clase Burkitt. Para la misma clase se obtiene una distancia media de 0,0893 y para las demás clases una distancia mínima de 0,1547.

Tabla 6.39. Resultados función oneshot para la clase Burkitt (Bloque 6).

Clase	Clase a comparar	Distancia	Distancia normalizada
BURKITT	BURKITT	0,0893	100%
	FOLICULAR	0,1547	173%
	HCL	0,2371	266%
	LCM	0,1622	182%
	LLC	0,1942	217%

En la Tabla 6.40 se observan los resultados de aplicar la función oneshot para la clase Folicular. Para la misma clase se obtiene una distancia de 0,0934 y para las demás clases una distancia mínima de 0,1189.

Tabla 6.40. Resultados función oneshot para la clase Folicular (Bloque 6).

Clase	Clase a comparar	Distancia	Distancia normalizada
FOLICULAR	BURKITT	0,2857	306%
	FOLICULAR	0,0934	100%
	HCL	0,1370	147%
	LCM	0,1475	158%
	LLC	0,1189	127%

En la Tabla 6.41 se observan los resultados de aplicar la función oneshot para la clase HCL. Para la misma clase se obtiene una distancia media de 0,0417 y para las demás clases una distancia mínima de 0,0549.

Tabla 6.41. Resultados función oneshot para la clase HCL (Bloque 6).

Clase	Clase a comparar	Distancia	Distancia normalizada
HCL	BURKITT	0,2371	569%
	FOLICULAR	0,1303	312%
	HCL	0,0417	100%
	LCM	0,0831	199%
	LLC	0,0549	132%

En la Tabla 6.42 se observan los resultados de aplicar la función oneshot para la clase LCM. Para la misma clase se obtiene una distancia media de 0,1480 y para las demás clases una distancia mínima de 0,0626.

Tabla 6.42. Resultados función oneshot para la clase LCM (Bloque 6).

Clase	Clase a comparar	Distancia	Distancia normalizada
LCM	BURKITT	0,1570	106%
	FOLICULAR	0,3292	222%
	HCL	0,3561	241%
	LCM	0,1480	100%
	LLC	0,0626	42%

En la Tabla 6.43 se observan los resultados de aplicar la función oneshot para la clase LLC. Para la misma clase se obtiene una distancia media de 0,1006 y para las demás clases una distancia mínima de 0,0946.

Tabla 6.43. Resultados función oneshot para la clase LLC (Bloque 6).

Clase	Clase a comparar	Distancia	Distancia normalizada
LLC	BURKITT	0,1942	193%
	FOLICULAR	0,2811	279%
	HCL	0,0946	94%
	LCM	0,1778	177%
	LLC	0,1006	100%

6.8. Comparativa SNN (ResNet34 y ResNet50) con y sin pre-entrenamiento

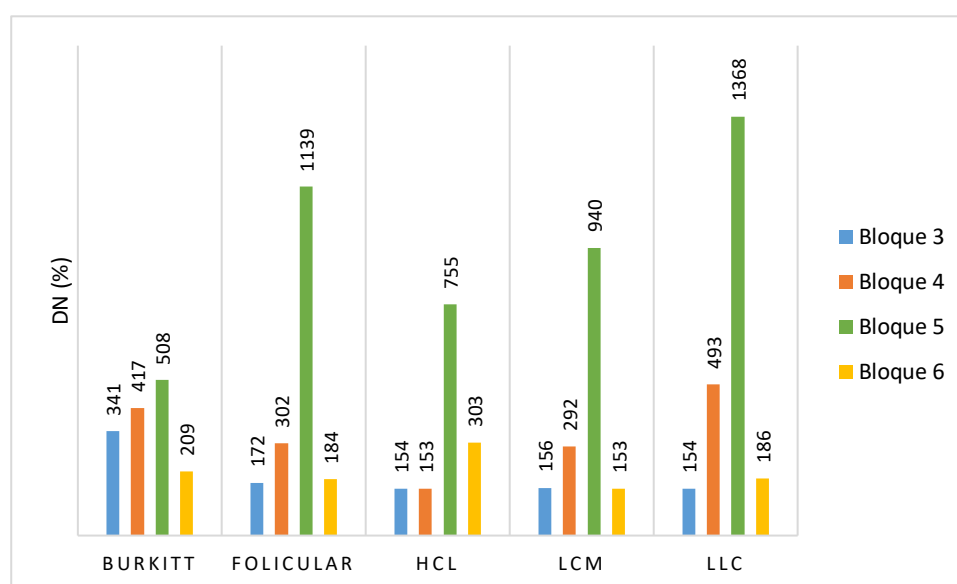
En este capítulo se comparan los resultados de utilizar SNN con arquitecturas ResNet34 y ResNet50 con y sin pre-entrenamiento para la clasificación de las cinco clases de linfomas. En la Tabla 6.44 se observan los resultados de aplicar la función oneshot para los bloques 3, 4, 5 y 6, donde:

- D: distancia.
- D_N : promedio de las distancias normalizadas para cada clase.

Tabla 6.44. Comparativa de los resultados obtenidos con la función oneshot de los bloques 3, 4, 5 y 6.

	Bloque 3		Bloque 4		Bloque 5		Bloque 6	
	D	D _N	D	D _N	D	D _N	D	D _N
BURKITT	0,4978	341%	0,0718	417%	0,0003	508%	0,0893	209%
FOLICULAR	1,0534	172%	0,0471	302%	$6,1482 \times 10^{-5}$	1.139%	0,0934	184%
HCL	1,0427	154%	0,092	153%	$9,5990 \times 10^{-5}$	755%	0,0417	303%
LCM	0,9128	156%	0,0432	292%	$8,2485 \times 10^{-5}$	940%	0,1480	153%
LLC	0,8908	154%	0,0210	493%	$6,3971 \times 10^{-5}$	1.368%	0,1006	186%

Para una mejor visualización, en la Figura 6.12 se recogen las distancias promediadas (D_N) de la Tabla 6.44.

**Figura 6.12.** Comparativa de las distancias normalizadas (D_N) de los bloques 3, 4, 5 y 6 para las cinco clases de linfomas.

En la Figura 6.12 se observa que el modelo que da mejores resultados para cualquiera de las clases es el bloque 5, es decir, una SNN con arquitectura ResNet34 pre-entrenada.

Para la clase Burkitt se obtiene el mejor resultado con el bloque 5 con una D_N de 508%. El peor resultado se obtiene con el bloque 6 con una D_N de 209%.

Para la clase Folicular se obtiene el mejor resultado con el bloque 5 con una D_N de 1.139%. El peor resultado se obtiene con el bloque 3 con una D_N de 172%.

Para la clase HCL se obtiene el mejor resultado con el bloque 5 con una D_N de 755%. El peor resultado se obtiene con el bloque 4 con una D_N de 153%.

Para la clase LCM se obtiene el mejor resultado con el bloque 5 con una D_N de 940%. El peor resultado se obtiene con el bloque 6 con una D_N de 153%.

Para la clase LLC se obtiene el mejor resultado con el bloque 5 con una D_N de 1.368%. El peor resultado se obtiene con el bloque 3 con una D_N de 154%.

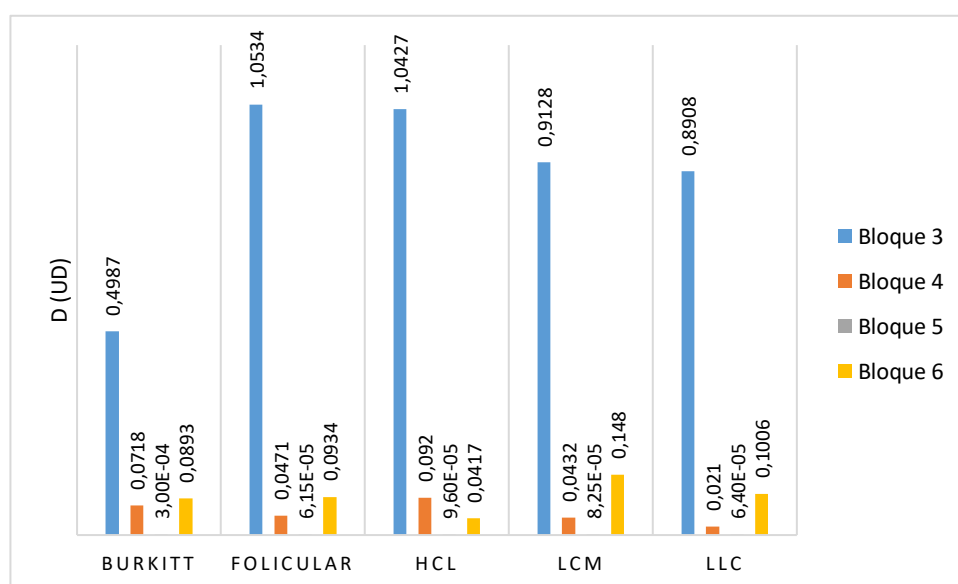


Figura 6.13. Comparativa de las distancias (D) de las respectivas clases de los bloques 3, 4, 5 y 6.

La Figura 6.13 sirve para ilustrar que para evaluar el desempeño de los modelos no es suficiente con ver cómo diferencian las clases entre sí. Es necesario también que el modelo sepa encontrar similitudes entre imágenes de la misma clase, es por eso que el valor distancia (D) sea lo menor posible. Se puede observar como el modelo del bloque 5 es el que mejor relaciona imágenes de la misma clase por un margen muy amplio. Aunque en algunas clases el modelo del bloque 3 presente un mejor desempeño que el modelo del bloque 6, la Figura 6.13 muestra como el modelo del bloque 3 no encuentra bien las similitudes entre imágenes de la misma clase.

7. Análisis del impacto ambiental

Tal y como se comenta en el capítulo 9.1, no se cuentan como gastos económicos el microscopio necesario para la toma de imágenes ni la realización de los frotis sanguíneos; por lo que no se tendrá en cuenta su impacto ambiental ya que las imágenes fueron proporcionadas de antemano por el Hospital Clínic.

Cabe destacar también que si algún componente hardware se avería y debe ser sustituido es necesario deshacerse de estos siguiendo el Real Decreto 110/2015 sobre residuos de aparatos eléctricos y electrónicos (99), puesto que estos dispositivos contienen elementos dañinos para el medio ambiente si no son gestionados correctamente.

Aunque este proyecto no requiere de grandes recursos y su impacto ambiental es muy bajo, debe tenerse en cuenta el consumo eléctrico del ordenador ya que supone un impacto ambiental no despreciable, explicado más detalladamente en el capítulo 9.4.



8. Conclusiones y extensiones futuras

8.1. Conclusiones

En lo que respecta a los dos primeros bloques de metodología de este proyecto (secciones 5.3.1 y 5.3.2) se ha realizado satisfactoriamente una clasificación de las imágenes de linfocitos. Se ha conseguido una *accuracy* del 93% utilizando una ResNet34 (Bloque 1) y una *accuracy* del 98% utilizando una ResNet50 (Bloque 2). Aunque el coste computacional es algo más elevado, se recomienda el uso de la ResNet50 por encima de la ResNet34 para realizar clasificaciones convencionales.

Mediante los modelos de los bloques 3 (SNN con arquitectura ResNet34 sin pre-entrenamiento) y 4 (SNN con arquitectura ResNet50 sin pre-entrenamiento) se ha observado una pequeña mejora en la diferenciación de imágenes al utilizar una ResNet50. Ahora bien, el modelo del bloque 3 ha sido el que ha presentado el peor desempeño en cuanto a encontrar similitudes entre imágenes de la misma clase.

Comparando los modelos SNN con arquitectura ResNet34 sin y con pre-entrenamiento; se concluye que el uso de los pesos obtenidos en el bloque 1 (ResNet34 convencional) permite aumentar en gran medida el desempeño del modelo del bloque 5 (SNN con arquitectura ResNet34 con pre-entrenamiento). De hecho, el modelo de SNN con arquitectura ResNet34 con pre-entrenamiento presenta los mejores valores de similitud entre imágenes de la misma clase.

Si se comparan los modelos SNN con arquitectura ResNet50 sin y con pre-entrenamiento; se observa que el desempeño del modelo ha empeorado tanto en encontrar diferencias y similitudes de las imágenes. Cabe destacar que el modelo del bloque 6 (SNN con arquitectura ResNet50 con pre-entrenamiento) no diferencia correctamente entre las clases HCL, LCM y LLC (se encuentran valores de D_N inferiores para las demás clases que para la clase en sí).

Comparando los resultados obtenidos de los modelos SNN; se determina que el uso del modelo SNN con arquitectura ResNet34 con pre-entrenamiento es el más adecuado. Se ha comprobado que el modelo presenta unos resultados excelentes para encontrar similitudes entre imágenes de la misma clase y diferencias entre imágenes de clases distintas.

Aunque el modelo ResNet50 convencional (bloque 2) haya presentado resultados excelentes, solamente es recomendable su uso para clasificaciones convencionales de estas cinco clases. En caso de que se añadiese una imagen que perteneciese a otro tipo de linfoma, el modelo se vería obligado a clasificarlo como una de las cinco clases con las que ha sido entrenado.

Par solventar este problema se recomienda el uso del modelo SNN con arquitectura ResNet34 con pre-entrenamiento (bloque 5). Puesto que no es un modelo basado en clasificaciones, al introducir una imagen no perteneciente a ninguna de las cinco clases, se obtendrían unas distancias muy elevadas para las cinco clases, con lo que se podría determinar que el nuevo linfoma no pertenece a ninguna de estas clases. De esta manera se descarta la posibilidad de un falso diagnóstico.

8.2. Extensiones futuras

En la Figura 6.2 se observa que los errores de clasificación más comunes del modelo son debidos a las células de alrededor de la imagen y no al linfocito en sí. Es por eso que se recomienda como una posible extensión futura del proyecto realizar un modelo de pre-procesado de las imágenes que resalte más los linfocitos o recorte las demás células. Esto permitiría que el modelo se viese obligado a aprender de la información que resulta más importante para la clasificación de las imágenes.

En el caso de que un modelo de clasificación basado en SNN llegase a producción, se aconseja la periódica comprobación de la calidad de sus clasificaciones mediante el re-entrenamiento del modelo. Es importante destacar también que, a medida que aumente la base de datos disponible de las distintas patologías, debe estudiarse hasta qué punto debe afrontarse el problema con SNN o si deberían utilizarse otros tipos de clasificación de imágenes donde el factor determinante no es el FSL. Se espera que estas cuestiones puedan evaluarse experimentalmente en el laboratorio del Hospital Clínic de Barcelona en un futuro.

Pueden utilizarse las características obtenidas en la capa final de la CNN dentro de la SNN para desarrollar algoritmos de ML utilizando metodologías de PCA para crear clasificadores con la información obtenida de los modelos de las redes neuronales de este proyecto.

9. Análisis económico

En este capítulo se presenta detalladamente la evaluación económica del proyecto. Esta evaluación económica se divide entre costes directos y costes indirectos además de añadir un porcentaje de contingencia para una mayor precisión. Los costes directos del proyecto incluyen costes de hardware, software y costes de personal.

9.1. Costes de hardware

La mayoría de *hardware* usado en este proyecto es básico, puesto que se ha utilizado un simple ordenador portátil, a excepción del servidor proporcionado por la estación de trabajo *deepbox*, que debido a sus prestaciones se han conseguido unas velocidades de computación mucho más elevadas que si se hubiese usado el portátil solamente. Es por eso que la GPU de la estación de trabajo se contará como un gasto indispensable para la realización de este proyecto.

Debido a que las imágenes han sido directamente proporcionadas por el Hospital Clínic no se contarán como gasto el microscopio necesario para la toma de las imágenes ni la realización de los distintos frotis sanguíneos.

Tabla 9.1. Costes directos de hardware.

	Precio (€)
NVIDIA GeForce Titan Xp 12GB (100)	1.115,66
IVA (21%)	234,29
TOTAL	1.349,95

9.2. Costes de software

Los programas utilizados en este proyecto (tanto los programas de conexión VPN, las librerías de programación y entornos de trabajo) son de código abierto por lo que no incurren ningún coste.

9.3. Costes de personal

Como se ha comentado en el capítulo 9.1, no se han tenido en cuenta los costes del microscopio ni de los frotis sanguíneos; por lo que no se tendrán en cuenta los costes de mano de obra asociados a estos dos conceptos. Como costes relativos a la ingeniería, se computarán las horas trabajadas por el autor de este proyecto y las horas del director y co-director. Las horas del autor del proyecto se contarán como ingeniero junior a 6 €/h y las del director y co-director se contarán como 30 €/h.

Tabla 9.2. Costes directos de personal.

	Horas (h)	Precio/Hora (€/h)	Precio (€)
Autor	600	10	6.000
Director	10	30	300
Co-director	40	30	1.200
IVA (21%)			1.575
TOTAL			9.075

9.4. Costes indirectos

Los costes indirectos hacen referencia a esos costes externos al desarrollo del proyecto. El más importante de ellos, reflejado en la Tabla 9.3, es debido al consumo eléctrico del ordenador.

Tabla 9.3. Costes indirectos (101)(102).

	Unidades (kW/h)	Coste (€/kWh)	Horas (h)	Precio (€)
Electricidad	0,40	0,15	600	36
IVA (21%)				7,56
TOTAL				43,56

9.5. Presupuesto total

A continuación se especifica el presupuesto total que se ha destinado a la realización de este proyecto, teniendo en cuenta los distintos recursos mencionados en los capítulos anteriores.

Se ha escogido también un porcentaje de contingencia del 5% por posibles olvidos en la estimación del presupuesto del proyecto.

Tabla 9.4. Presupuesto total.

Tipo	Precio (€)
Hardware	1.349,95
Software	0
Personal	9.075
Indirectos	43,56
SUBTOTAL	10.468,51
Contingencia (5%)	523,43
TOTAL	10.991,94

Como se ve reflejado en la Tabla 9.4, se realiza una tasación de este proyecto por un valor de 10.991,94 €.

Bibliografía

1. Las cifras del cáncer en España. A: [en línea]. 2019. [Consulta: 15 diciembre 2019]. Disponible a: www.isciii.es/ISCIII/es/contenidos/fd-servicios-cientifico-tecnicos/fd-vigilancias-alertas/fd-epidemiologia-am-.
2. Definición de sangre - Diccionario de cáncer - National Cancer Institute. A: [en línea]. [Consulta: 16 diciembre 2019]. Disponible a: <https://www.cancer.gov/espanol/publicaciones/diccionario/def/sangre>.
3. Guyton, A. i Hall, J. *Tratado de fisiología médica*. 2006.
4. Sangre y hemopoyesis. A: [en línea]. [Consulta: 16 diciembre 2019]. Disponible a: <https://www.monografias.com/trabajos906/sangre-hemopoyesis-hematopoyesis/sangre-hemopoyesis-hematopoyesis2.shtml>.
5. Factores de crecimiento hematopoyético. A: [en línea]. [Consulta: 17 diciembre 2019]. Disponible a: https://www.franrzm.com/factores-de-crecimiento-hematopoyetico/#Interleucina-3_IL-3.
6. Producción de eritrocitos - Hematología y oncología - Manual MSD. A: [en línea]. [Consulta: 17 diciembre 2019]. Disponible a: <https://www.msdmanuals.com/es-es/professional/hematología-y-oncología/abordaje-del-paciente-con-anemia/producción-de-eritrocitos>.
7. Definición de hemoglobina - Diccionario de cáncer - National Cancer Institute. A: [en línea]. [Consulta: 30 diciembre 2019]. Disponible a: <https://www.cancer.gov/espanol/publicaciones/diccionario/def/hemoglobina>.
8. Introducción al sistema inmunitario - Trastornos inmunológicos - Manual MSD. A: [en línea]. [Consulta: 19 diciembre 2019]. Disponible a: https://www.msdmanuals.com/es-es/hogar/trastornos-inmunológicos/biología-del-sistema-inmunitario/introducción-al-sistema-inmunitario#v778515_es.
9. Introducción a los trastornos de los glóbulos blancos (leucocitos) - Trastornos de la sangre - Manual MSD. A: [en línea]. [Consulta: 31 diciembre 2019]. Disponible a: <https://www.msdmanuals.com/es-es/hogar/trastornos-de-la-sangre/trastornos-de-los-glóbulos-blancos-leucocitos/introducción-a-los-trastornos-de-los-glóbulos-blancos-leucocitos>.
10. Granulocito: MedlinePlus. A: [en línea]. [Consulta: 19 diciembre 2019]. Disponible a: https://medlineplus.gov/spanish/ency/esp_imagepages/9355.htm.
11. Definición y tipos de linfocitos, y sus correlaciones clínicas. A: [en línea]. [Consulta: 15 diciembre 2019]. Disponible a: <https://www.elsevier.com/es-es/connect/medicina/tipos-de-linfocitos-linfoma-Hodgkin>.
12. El sistema inmunitario: Información acerca de linfocitos, células dendríticas, macrófagos y leucocitos - Información de Quimioterapia. A: [en línea]. [Consulta: 15 diciembre 2019]. Disponible a: <http://chemocare.com/es/chemotherapy/what-is-chemotherapy/el-sistema-inmunitario.aspx>.
13. Componentes de la sangre - Manual MSD. A: [en línea]. [Consulta: 19 diciembre 2019]. Disponible a: <https://www.msdmanuals.com/es-es/hogar/trastornos-de-la-sangre/biología-de-la>

sangre/componentes-de-la-sangre.

14. Sistema linfático: MedlinePlus. A: [en línea]. [Consulta: 19 diciembre 2019]. Disponible a: <https://medlineplus.gov/spanish/ency/article/002247.htm>.
15. Sistema linfático - Diccionario de cáncer - National Cancer Institute. A: [en línea]. [Consulta: 19 diciembre 2019]. Disponible a: <https://www.cancer.gov/espanol/publicaciones/diccionario/def/sistema-linfatico>.
16. El sistema linfático | AEAL. A: [en línea]. [Consulta: 19 diciembre 2019]. Disponible a: <http://www.aeal.es/informacion-general-linfoma/3-el-sistema-linfatico/>.
17. Introducción al sistema linfático - Trastornos del corazón y los vasos sanguíneos - Manual MSD. A: [en línea]. [Consulta: 20 diciembre 2019]. Disponible a: <https://www.msmanuals.com/es/hogar/trastornos-del-corazon-y-los-vasos-sanguineos/trastornos-del-sistema-linfatico/introduccion-al-sistema-linfatico>.
18. Ganglio linfático no neoplásico | Patología | AccessMedicina | McGraw-Hill Medical. A: [en línea]. [Consulta: 8 gener 2020]. Disponible a: <https://accessmedicina.mhmedical.com/content.aspx?bookid=1493§ionid=102870723>.
19. Preparar un frotis de sangre. A: [en línea]. [Consulta: 20 diciembre 2019]. Disponible a: <http://www.vivo.colostate.edu/hatosano/topics/bloodsmear.html>.
20. Tinción de May Grünwald-Giemsa: fundamento, técnica y usos. A: [en línea]. [Consulta: 20 diciembre 2019]. Disponible a: <https://www.lifeder.com/tincion-may-grunwald-giemsa/>.
21. Importancia del estudio del frotis de sangre periférica en ancianos. A: [en línea]. [Consulta: 20 diciembre 2019]. Disponible a: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1727-897X2017000300012.
22. Definición de neutrófilo - Diccionario de cáncer - National Cancer Institute. A: [en línea]. [Consulta: 31 diciembre 2019]. Disponible a: <https://www.cancer.gov/espanol/publicaciones/diccionario/def/neutrofilo>.
23. Neutropenia - Trastornos de la sangre - Manual MSD. A: [en línea]. [Consulta: 7 gener 2020]. Disponible a: <https://www.msmanuals.com/es-es/hogar/trastornos-de-la-sangre/trastornos-de-los-globulos-blancos-leucocitos/neutropenia>.
24. Bonilla, M.A. i Menell, J.S. Disorders of White Blood Cells. A: [en línea]. 2016, DOI 10.1016/B978-0-12-801368-7.00013-2. [Consulta: 6 gener 2020]. Disponible a: <http://dx.doi.org/10.1016/B978-0-12-801368-7.00013-2>.
25. Leucocitosis neutrófila - Trastornos de la sangre - Manual MSD. A: [en línea]. [Consulta: 7 gener 2020]. Disponible a: <https://www.msmanuals.com/es-es/hogar/trastornos-de-la-sangre/trastornos-de-los-globulos-blancos-leucocitos/leucocitosis-neutrofila>.
26. Trastornos de los eosinófilos - Trastornos de la sangre - Manual MSD. A: [en línea]. [Consulta: 6 gener 2020]. Disponible a: <https://www.msmanuals.com/es-es/hogar/trastornos-de-la-sangre/trastornos-de-los-globulos-blancos-leucocitos/trastornos-de-los-eosinofilos>.

27. Definición de basófilo - Diccionario de cáncer - National Cancer Institute. A: [en línea]. [Consulta: 6 gener 2020]. Disponible a: <https://www.cancer.gov/espanol/publicaciones/diccionario/def/basofilo>.
28. Alergias (síntomas, sensibilización) y el papel de la histamina. A: [en línea]. [Consulta: 6 gener 2020]. Disponible a: <https://www.elsevier.com/es-es/connect/medicina/alergias-sintomas,-sensibilizacion-y-el-papel-de-la-histamina>.
29. Trastornos de los basófilos - Trastornos de la sangre - Manual MSD. A: [en línea]. [Consulta: 6 gener 2020]. Disponible a: <https://www.msdmanuals.com/es-es/hogar/trastornos-de-la-sangre/trastornos-de-los-glóbulos-blancos-leucocitos/trastornos-de-los-basófilos>.
30. Trastornos de los monocitos - Trastornos de la sangre - Manual MSD. A: [en línea]. [Consulta: 6 gener 2020]. Disponible a: <https://www.msdmanuals.com/es-es/hogar/trastornos-de-la-sangre/trastornos-de-los-glóbulos-blancos-leucocitos/trastornos-de-los-monocitos>.
31. Linfocitopenia - Trastornos de la sangre - Manual MSD. A: [en línea]. [Consulta: 7 gener 2020]. Disponible a: <https://www.msdmanuals.com/es-es/hogar/trastornos-de-la-sangre/trastornos-de-los-glóbulos-blancos-leucocitos/linfocitopenia>.
32. Linfocitopenia - Hematología y oncología - Manual MSD. A: [en línea]. [Consulta: 7 gener 2020]. Disponible a: <https://www.msdmanuals.com/es-es/professional/hematología-y-oncología/leucopenias/linfocitopenia>.
33. Leucocitosis linfocítica - Trastornos de la sangre - Manual MSD. A: [en línea]. [Consulta: 15 diciembre 2019]. Disponible a: <https://www.msdmanuals.com/es-es/hogar/trastornos-de-la-sangre/trastornos-de-los-glóbulos-blancos-leucocitos/leucocitosis-linfocítica>.
34. Linfoma - diagnóstico, evaluación y tratamiento. A: [en línea]. [Consulta: 27 diciembre 2019]. Disponible a: <https://www.radiologyinfo.org/sp/info.cfm?pg=lymphoma>.
35. Linfoma de Hodgkin - Trastornos de la sangre - Manual MSD. A: [en línea]. [Consulta: 15 diciembre 2019]. Disponible a: <https://www.msdmanuals.com/es-es/hogar/trastornos-de-la-sangre/linfomas/linfoma-de-hodgkin>.
36. WebPathology - Classical Hodgkin Lymphoma: Nodular Sclerosis (Reed-Sternberg Cell). A: [en línea]. Disponible a: <https://www.webpathology.com/image.asp?n=5&Case=388>.
37. Linfomas no hodgkinianos - Trastornos de la sangre - Manual MSD. A: [en línea]. [Consulta: 15 diciembre 2019]. Disponible a: <https://www.msdmanuals.com/es-es/hogar/trastornos-de-la-sangre/linfomas/linfomas-no-hodgkinianos>.
38. Tratamiento del linfoma no Hodgkin en adultos - Instituto Nacional del Cáncer. A: [en línea]. [Consulta: 27 diciembre 2019]. Disponible a: <https://www.cancer.gov/espanol/tipos/linfoma/paciente/tratamiento-lnh-adultos-pdq>.
39. Definición de linfoma de Burkitt - Diccionario de cáncer - National Cancer Institute. A: [en línea]. [Consulta: 15 diciembre 2019]. Disponible a: <https://www.cancer.gov/espanol/publicaciones/diccionario/def/linfoma-de-burkitt>.
40. Linfoma de Burkitt - Hematología y oncología - Manual MSD versión para profesionales. A: [en línea].

[Consulta: 15 diciembre 2019]. Disponible a: <https://www.msdmanuals.com/es-es/professional/hematología-y-oncología/linfomas/linfoma-de-burkitt>.

41. Orphanet: Leucemia de células pilosas clásica. A: [en línea]. [Consulta: 15 diciembre 2019]. Disponible a: https://www.orpha.net/consor/cgi-bin/OC_Exp.php?Lng=ES&Expert=58017.

42. Troussard, X. Hairy cell leukemia 2018: Update on diagnosis, risk-stratification, and treatment. A: . DOI 10.1002/ajh.24936.

43. Esplenomegalia - Hematología y oncología - Manual MSD versión para profesionales. A: [en línea]. [Consulta: 8 gener 2020]. Disponible a: <https://www.msdmanuals.com/es-es/professional/hematología-y-oncología/trastornos-esplénicos/esplenomegalia>.

44. Leucemia de células pilosas | Genetic and Rare Diseases Information Center (GARD) – an NCATS Program. A: [en línea]. [Consulta: 8 gener 2020]. Disponible a: <https://rarediseases.info.nih.gov/espanol/13464/leucemia-de-celulas-pilosas>.

45. Información sobre el linfoma de células del manto. A: [en línea]. [Consulta: 8 gener 2020]. Disponible a: www.LLS.org/espanol.

46. Orphanet: Linfoma de células del manto. A: [en línea]. [Consulta: 8 gener 2020]. Disponible a: https://www.orpha.net/consor/cgi-bin/OC_Exp.php?Expert=52416&Lng=ES.

47. Adenopatía: definición- localización- orígenes y causas de esta inflamación. A: [en línea]. [Consulta: 30 diciembre 2019]. Disponible a: <https://www.elsevier.com/es-es/connect/medicina/adenopatia-definicion-localizacion-origenes-y-causas-de-esta-inflamacion>.

48. 3. El linfoma de células del manto | AEAL. A: [en línea]. [Consulta: 8 gener 2020]. Disponible a: <http://www.aeal.es/linfoma-de-celulas-del-manto-espana/3-el-linfoma-de-celulas-del-manto/>.

49. Orphanet: Linfoma folicular. A: [en línea]. [Consulta: 30 diciembre 2019]. Disponible a: https://www.orpha.net/consor/cgi-bin/OC_Exp.php?Lng=ES&Expert=545.

50. Definición de linfoma folicular - Diccionario de cáncer - National Cancer Institute. A: [en línea]. [Consulta: 30 diciembre 2019]. Disponible a: <https://www.cancer.gov/espanol/publicaciones/diccionario/def/linfoma-folicular>.

51. McNamara, C. et al. Guidelines on the investigation and management of follicular lymphoma. A: *British Journal of Haematology* [en línea]. 2012, Vol. 156, núm. 4, p. 446-467. DOI 10.1111/j.1365-2141.2011.08969.x. Disponible a: <http://doi.wiley.com/10.1111/j.1365-2141.2011.08969.x>.

52. ENFERMEDADES MALIGNAS DE LA SANGRE. A: [en línea]. [Consulta: 8 gener 2020]. Disponible a: <http://atlasgeneticsoncology.org/Educ/HempatID30050SS.html>.

53. Turbatu, A. et al. Composite Diffuse Large B-Cell Lymphoma and Follicular B-Cell Lymphoma-Case Report and Review of Literature. A: *Journal of Clinical Medicine CASE REPORTS 204 Maedica A Journal of Clinical Medicine*. 2014.

54. Leucemia linfocítica crónica - Trastornos de la sangre - Manual MSD. A: [en línea]. [Consulta: 29 diciembre 2019]. Disponible a: <https://www.msdmanuals.com/es-es/hogar/trastornos-de-la>



sangre/leucemias/leucemia-linfocítica-crónica.

55. Síndrome de Richter - Diccionario de cáncer - National Cancer Institute. A: [en línea]. [Consulta: 29
diciembre 2019]. Disponible a: <https://www.cancer.gov/espanol/publicaciones/diccionario/def/sindrome-de-richter>.

56. Leucemia linfocítica crónica: MedlinePlus. A: [en línea]. [Consulta: 29 diciembre 2019]. Disponible
a: <https://medlineplus.gov/spanish/chroniclymphocyticleukemia.html>.

57. Russell, S. i Norvig, P. *Inteligencia Artificial - Un enfoque Moderno* [en línea]. ISBN 978-84-205-4003-
0. [Consulta: 3 març 2020]. Disponible a: www.pearsoneducacion.com.

58. Everything you need to know about Neural Networks and Backpropagation. A: [en línea]. [Consulta:
23 diciembre 2019]. Disponible a: <https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a>.

59. Chokmani, K. et al. Estimation of River Ice Thickness Using Artificial Neural Networks. A: *Proceedings
- 14th Workshop on the Hydraulics of Ice Covered Rivers, Quebec City, CGU-HS CRIPE*. 2007, núm. July
2014, p. 1-12.

60. Role of Bias in Neural Networks - Intellipaat Community. A: [en línea]. [Consulta: 3 març 2020].
Disponible a: <https://intellipaat.com/community/253/role-of-bias-in-neural-networks>.

61. Rumelhart, D.E., Hinton, G.E. i Williams, R.J. Learning representations by back-propagating errors.
A: *Nature*. 1986, Vol. 323.

62. Krizhevsky, A., Sutskever, I. i Hinton, G.E. ImageNet Classification with Deep Convolutional Neural
Networks. A: [en línea]. [Consulta: 3 març 2020]. Disponible a: <http://code.google.com/p/cuda-convnet/>.

63. Activation Functions in Neural Networks - Towards Data Science. A: [en línea]. [Consulta: 3 març
2020]. Disponible a: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>.

64. Understanding different Loss Functions for Neural Networks. A: [en línea]. [Consulta: 10 març 2020].
Disponible a: <https://towardsdatascience.com/understanding-different-loss-functions-for-neural-networks-dd1ed0274718>.

65. Understanding Confusion Matrix - Towards Data Science. A: [en línea]. [Consulta: 11 març 2020].
Disponible a: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>.

66. Confusion Matrix - Applied Deep Learning with Keras. A: [en línea]. [Consulta: 27 abril 2020].
Disponible a: https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781838555078/6/ch06lvl1sec34/confusion-matrix.

67. What Are Overfitting and Underfitting in Machine Learning? A: [en línea]. [Consulta: 10 març 2020].
Disponible a: <https://towardsdatascience.com/what-are-overfitting-and-underfitting-in-machine-learning-a96b30864690>.

68. Underfitting and Overfitting in Machine Learning - GeeksforGeeks. A: [en línia]. [Consulta: 10 març 2020]. Disponible a: <https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/>.
69. Overfitting y underfitting | de-aprendizaje-de-la-máquina. A: [en línia]. [Consulta: 10 març 2020]. Disponible a: <https://www.i-ciencias.com/pregunta/134998/overfitting-y-underfitting>.
70. Fully Connected Layers in Convolutional Neural Networks: The Complete Guide - MissingLink.ai. A: [en línia]. [Consulta: 10 març 2020]. Disponible a: <https://missinglink.ai/guides/convolutional-neural-networks/fully-connected-layers-convolutional-neural-networks-complete-guide/>.
71. Jönsson, J. i Sjöstrand, E. Cell Image Transformation Using Deep Learning. A: *LUND UNIVERSITY LIBRARIES*. 2018.
72. A Gentle Introduction to Pooling Layers for Convolutional Neural Networks. A: [en línia]. [Consulta: 10 març 2020]. Disponible a: <https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>.
73. The Vanishing Gradient Problem - Anish Singh Walia - Medium. A: [en línia]. [Consulta: 3 març 2020]. Disponible a: <https://medium.com/@anishsingh20/the-vanishing-gradient-problem-48ae7f501257>.
74. He, K. et al. Deep Residual Learning for Image Recognition. A: [en línia]. [Consulta: 9 març 2020]. Disponible a: <http://image-net.org/challenges/LSVRC/2015/>.
75. An Overview of ResNet and its Variants - Towards Data Science. A: [en línia]. [Consulta: 3 març 2020]. Disponible a: <https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035>.
76. Understanding and visualizing ResNets - Towards Data Science. A: [en línia]. [Consulta: 5 març 2020]. Disponible a: <https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8>.
77. Wang, Y. et al. Generalizing from a Few Examples: A Survey on Few-Shot Learning. A: [en línia]. 2019, Disponible a: <http://arxiv.org/abs/1904.05046>.
78. Bromley, J. et al. Signature Verification using a «Siamese» Time Delay Neural Network. A: *International Journal of Pattern Recognition and Artificial Intelligence*. 1993.
79. Rao, S.J. i Cottrell, G.W. A Deep Siamese Neural Network Learns the Human-Perceived Similarity Structure of Facial Expressions Without Explicit Categories. A: *COGSI*. 2011, p. 217-222.
80. Siamese Networks: Algorithm, Applications And PyTorch Implementation. A: [en línia]. [Consulta: 15 desembre 2019]. Disponible a: <https://becominghuman.ai/siamese-networks-algorithm-applications-and-pytorch-implementation-4ffa3304c18>.
81. SP-1000 Side Maker Stainer. A: [en línia]. [Consulta: 18 febrer 2020]. Disponible a: <https://www.sysmex.com/la/es/Products/Hematology/Sysmex-Hematology-Automation/Pages/SP-1000-Side-Maker-Stainer.aspx>.

82. Riley, Roger S, Ben-Ezra, Jonathan M, Massey, Davis, C.J. The virtual blood film. A: *Clinics in Laboratory Medicine*. 2002, Vol. 22, núm. 1, p. 317-345.
83. CellaVision DM96. A: [en línia]. [Consulta: 18 febrer 2020]. Disponible a: https://www.sysmex.com/us/en/Products/Hematology/CellImageAnalysis/Documents/Brochure_DM96.pdf.
84. CellaVision DM96 Digital Cell Morphology System - Siemens Healthineers. A: [en línia]. [Consulta: 19 febrer 2020]. Disponible a: <https://www.siemens-healthineers.com/nl-be/hematology/systems/cellavision-dm96-digital-cell-morphology-system>.
85. vision.learner | fastai. A: [en línia]. [Consulta: 10 març 2020]. Disponible a: <https://docs.fast.ai/vision.learner.html>.
86. basic_train | fastai. A: [en línia]. [Consulta: 10 març 2020]. Disponible a: https://docs.fast.ai/basic_train.html#fit_one_cycle.
87. Finding Good Learning Rate and The One Cycle Policy. A: [en línia]. [Consulta: 10 març 2020]. Disponible a: <https://towardsdatascience.com/finding-good-learning-rate-and-the-one-cycle-policy-7159fe1db5d6>.
88. Smith, L.N. i Topin, N. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. A: *Cornell University*. 2018.
89. vision.learner | fastai. A: [en línia]. [Consulta: 10 març 2020]. Disponible a: <https://docs.fast.ai/vision.learner.html#ClassificationInterpretation>.
90. basic_train | fastai. A: [en línia]. [Consulta: 10 març 2020]. Disponible a: https://docs.fast.ai/basic_train.html#Learner.unfreeze.
91. callbacks.lr_finder | fastai. A: [en línia]. [Consulta: 10 març 2020]. Disponible a: https://docs.fast.ai/callbacks.lr_finder.html.
92. Understanding Learning Rates and How It Improves Performance in Deep Learning. A: [en línia]. [Consulta: 10 març 2020]. Disponible a: <https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10>.
93. basic_train | fastai. A: [en línia]. [Consulta: 11 març 2020]. Disponible a: https://docs.fast.ai/basic_train.html#Learner.show_results.
94. vision.data | fastai. A: [en línia]. [Consulta: 18 març 2020]. Disponible a: <https://docs.fast.ai/vision.data.html#ImageDataBunch>.
95. vision.data | fastai. A: [en línia]. [Consulta: 18 març 2020]. Disponible a: https://docs.fast.ai/vision.data.html#ImageDataBunch.from_name_re.
96. core | fastai. A: [en línia]. [Consulta: 18 març 2020]. Disponible a: <https://docs.fast.ai/core.html#ItemBase>.
97. Contrastive Loss Explained - Towards Data Science. A: [en línia]. [Consulta: 18 març 2020].

Disponible a: <https://towardsdatascience.com/contrastive-loss-explained-159f2d4a87ec>.

98. training | fastai. A: [en línea]. [Consulta: 26 març 2020]. Disponible a: <https://docs.fast.ai/training.html>.

99. *Real Decreto 110/2015, de 20 de febrero, sobre residuos de aparatos eléctricos y electrónicos.*

100. NVIDIA GeForce® Titan Xp Pascal 12GB GDDR5X - Tarjeta Gráfica. A: [en línea]. [Consulta: 27 febrer 2020]. Disponible a: <https://www.coolmod.com/nvidia-geforce-titan-xp-pascal-12gb-gddr5x-tarjeta-grafica-precio>.

101. ¿Cuánto consume realmente nuestro ordenador? A: [en línea]. [Consulta: 26 febrer 2020]. Disponible a: <https://hardzone.es/cuanto-consume-realmente-nuestro-ordenador/>.

102. Precio del kwh en España: Comparativa de Endesa, Iberdrola y otras. A: [en línea]. [Consulta: 26 febrer 2020]. Disponible a: <https://tarifaluzhora.es/info/precio-kwh#precio-kwh-españa>.

